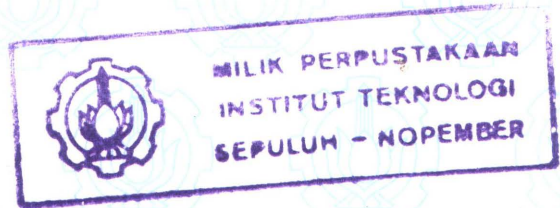


22266/H/05



DESAIN DAN IMPLEMENTASI PERANGKAT LUNAK
PENGUKURAN KINERJA MANAJEMEN STRATEGIS PERUSAHAAN
MENGGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS:
STUDI KASUS DI PT MERPATI NUSANTARA AIRLINES SURABAYA

TUGAS AKHIR



RSIF
005.1
Kam
d-1
2004

Disusun oleh :
MUSTAFA KAMAL
5100 100 018

PERPUSTAKAAN ITS	
Tgl. Terima	11-8-2004
Terima Dari	A1
No. Agenda Prp.	220842

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

2004

**DESAIN DAN IMPLEMENTASI PERANGKAT LUNAK
PENGUKURAN KINERJA MANAJEMEN STRATEGIS PERUSAHAAN
MENGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS:
STUDI KASUS DI PT MERPATI NUSANTARA AIRLINES SURABAYA**

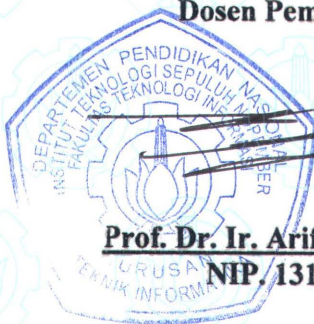
TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Komputer
Pada**

**Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui/Menyetujui

Dosen Pembimbing I,



**Prof. Dr. Ir. Arif Djunaidy, M.Sc
NIP. 131 633 403**

Dosen Pembimbing II,



**Darlis Heru Murti, S.Kom
NIP. 132 306 430**

SURABAYA

Juli 2004

*Seberapa tinggi aku mendaki karier,
Seberapa luas aku meraih cakrawala,
Sebanyak apapun bintang yang kugenggam,
atau.. bahkan ketika aku menjadi raja maha diraja dan aku kuasai seluruh jagat dunia sekalipun,
Selalu ada batas yang tak bisa dilewati..*

*Ini bukan batas yang menghentikan langkah,
tapi batas untuk menyesuaikan makna hidup ini...*

Anda pun dapat melangkah lebih ringan.

*Bila selama ini anda mengumpulkan sesuatu,
maka batas itu menuntun anda untuk merelakannya lepas dari genggamannya.*

Sadarilah kapan saatnya batas itu tiba..

*Meski ada jutaan dahan kayu di hutan,
seekor Pipit cukup membutuhkan sebatang ranting untuk menggayutkan sarangnya.*

Keselamatan telur-telur mereka lebih berharga dibanding seberapa banyak ranting yang bisa dikuasai.

Bagaimana anda melihat kata "cukup" ?

*Apakah penting bagi anda memiliki kue sebanyak-banyaknya ?
Padahal anda takkan bisa menelan lebih besar dari lubang tenggorokan...*

Selalu ada batas untuk hal-hal yang anda miliki,

Namun, tiada kata cukup bila anda berkenan memberi...

hanya memberi,

dan memberi....

Itulah mengapa para serdadu sejati bersedia menjadi martir di medan tempur..

atau para mujahid menjadi syahid...

— Renungan : Mustafa Kamal —

ABSTRAK

Tingkat persaingan bisnis yang semakin kompetitif dalam era global saat ini telah membuat suatu perusahaan harus benar-benar dapat berkompetisi dengan perusahaan pesaingnya. Dalam memenangkan persaingan global tersebut, perusahaan sebagai organisasi bisnis, memiliki visi, misi dan tujuan yang pada implementasinya dijabarkan dalam sasaran-sasaran perusahaan. Berhasil tidaknya pencapaian sasaran-sasaran tersebut harus dipantau. Parameter-parameter pengukuran kinerja utama (*Key Performance Indicators/KPI*) perusahaan harus dipantau guna memenuhi target sasaran yang harus dicapai pada suatu periode tertentu. *Analytical Hierarchy Process* (AHP) merupakan salah satu metode pembobotan yang dapat digunakan untuk memantau perkembangan pencapaian kinerja perusahaan tersebut.

Dalam Tugas Akhir ini, dibuat perangkat lunak pengukuran kinerja manajemen strategis perusahaan menggunakan metode AHP. Implementasi perangkat lunak ini didasarkan pada suatu studi kasus yang dilakukan di PT Merpati Nusantara Airlines Surabaya. Perangkat lunak menerima masukan yang berupa data periode pengukuran kinerja, KPI, perbandingan matriks AHP, bobot KPI, target KPI dan pencapaian KPI. Sejumlah masukan ini digunakan oleh perangkat lunak untuk melakukan pengukuran kinerja perusahaan berdasarkan metode pembobotan AHP. Luaran dari perangkat lunak ini berupa data hasil pengukuran kinerja dalam bentuk tabel hasil pengukuran dan grafik perkembangan kinerja.

Berdasarkan hasil uji coba perangkat lunak, simulasi pencapaian target sasaran di PT Merpati Nusantara Airlines Surabaya akan dapat dilakukan dengan mudah, cepat dan akurat. Dengan demikian, pihak manajemen dapat melakukan tindakan koreksi, yang antara lain dapat berupa kebijakan manajemen strategis untuk periode berikutnya guna meningkatkan daya saing perusahaan.

KATA PENGANTAR

Tiada kata terindah yang dapat terlantun, *Alhamdulillah* rabbi 'alamin, segala puji bagi Allah Yang Maha Kuasa yang telah memberikan cinta-Nya sehingga penulis bisa menyelesaikan tugas akhir yang berjudul : **“DESAIN DAN IMPLEMENTASI PERANGKAT LUNAK PENGUKURAN KINERJA MANAJEMEN STRATEGIS PERUSAHAAN MENGGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS: STUDI KASUS DI PT MERPATI NUSANTARA AIRLINES SURABAYA”**

Pada tugas akhir ini, dibuat sebuah perangkat lunak yang dapat membantu pihak manajemen perusahaan dalam melakukan pengukuran kinerja. Penggunaan metode *Analytical Hierarchy Process* (AHP) yang dipilih dapat mempermudah pembobotan kriteria.

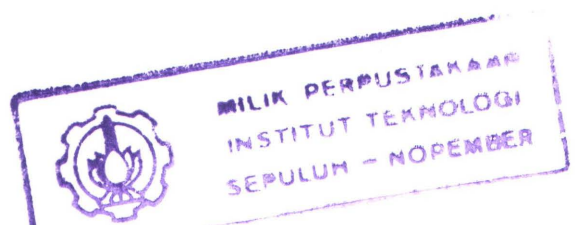
Penulis juga menyadari bahwa tugas akhir ini masing memiliki banyak kelemahan dan kekurangan. Oleh karena itu penulis sangat mengharapkan segala masukan bagi pengembangan selanjutnya.

Akhirul kalam, penulis mengucapkan syukur dan terima kasih kepada semua pihak yang telah membantu penyelesaian tugas akhir ini. Semoga Allah membalasnya dengan balasan sebaik-baiknya. Amien.

Salam perjuangan!

Surabaya, Juli 2004

Penulis



UCAPAN TERIMA KASIH

Bismillaahi Allahu Akbar. Alhamdulillah, puji syukur kehadiran Allah *Subhanahu Wa Ta'ala* yang senantiasa mencurahkan cinta-Nya. Tiada daya dan kekuatan melainkan dengan izin-Nya.

Secara khusus, dalam kesempatan ini pula, penulis ingin mengucapkan terima kasih dan penghargaan sebesar-besarnya kepada semua pihak yang telah menyertai dan membantu penulis selama ini, antara lain kepada:

1. Bapak Prof. Dr. Ir. Arif Djunaidy, M.Sc, selaku Ketua Dekan Fakultas Teknologi Informasi.
2. Bapak Yudhi Purwananto, M.Kom, selaku Ketua Jurusan Teknik Informatika.
3. Ibu Esther Hanaya, M.Sc, selaku Dosen wali penulis.
4. Bapak Prof. Dr. Ir. Arif Djunaidy, M.Sc, dan Bapak Darlis Heru Murti, S.Kom, selaku Dosen pembimbing penulis. *Terima kasih buanyak yaa Pak... Mohon maaf (10x) karena termasuk mahasiswa tugas akhir yang jarang bimbingan... ☺*
5. Bapak Dr. Ir. Joko Lianto B, Ibu Esther Hanaya, M.Sc dan Ibu Diana Purwitasari, S.Kom sebagai dosen penguji. *Cuman dikasih nilai B yaa Pak Bu... ☺ Makasih banyak atas ujiannya...*
6. Segenap dosen pengajar dan karyawan Fakultas Teknologi Informasi.
7. Papaku Masrur nan bijak dan Mamaku tercinta, Siti Fatimatul Zahro selaku orang tua penulis. *I love you mom dad.. Kebijakanmu sebagai tauladanku, Senyumanmu sebagai energiku, Cinta dan belai kasih sayangmu menyejukkan*

gelora jiwaku, Do'amu membuatku bagai bisa berlari secepat kilat. Andaikan aku bisa menggapai rembulan dan kupersembahkan padamu, maka jasamu takkan pernah tergantikan. *Aku ingin mendampingimu di surga-Nya, kelak...*

8. Mar'atus Solihah, Meshallina dan Nana Alfiana, selaku adik-adik kandung penulis. *Makasih udah ngasih semangat... Terimakasih atas telepon canda manjamu.. Setelah sekian lama sibuk TA, sungguh rindu rasanya untuk menjaga dan memanjakanmu... Semoga kakakmu ini bisa menjadi tauladan...*

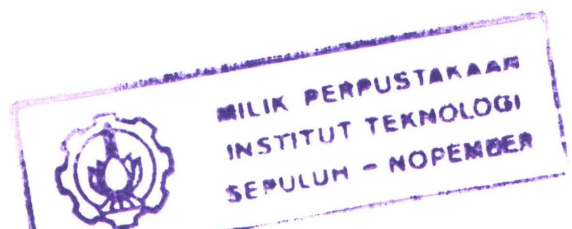
😊

9. Segenap keluarga dan kerabat tercinta, di Jakarta, Bandung, Bojonegoro dan Lamongan, serta kerabat lainnya yang tidak bisa disebutkan satu persatu.
10. Bapak Marlo Rumangkang selaku Manager HRD PT Merpati Nusantara, Bapak Agus Hartono dan Bapak Lakero selaku pembimbing lapangan Tugas Akhir.
11. Segenap pimpinan dan karyawan Divisi MMF, PT Merpati Nusantara Airlines Surabaya.
12. Seseorang yang sangat istimewa. *Kuncup mekarmu menjadi bunga... mempesona indah harum semerbak menawan hati...*
13. Mas Kendy (*Three in One*), Mas Fahmy, Haikal (*harusnya header+commentsnya dihapus yaa..😊*), Jakka (*3 hours..😊*), Jepe (*Struts..😊*), Mbak Ridha (*Suhu Rational Rose..😊*), yang telah banyak membantu pengerjaan Tugas Akhir. *Harusnya aku bagi-bagi nilai TA ama kalian yaa... hehehe...😊*

14. Saudara seperjuangan di Lab SI, Mas Budinug, Udin, Warna, Cdkey
seperjuangan dalam TA. Hendramania, Fauzan, Angga, Mumud seperjuangan
dalam bercanda... ☺
15. Kakak-kakak angkatanku tersayang... *Makasih kakak..*
16. Teman-teman noceng, angkatan millennium, angkatan yang paling kompak..
☺ *Kenangan manis senantiasa berlanjut..*
17. Adik-adik angkatanku yang imut-imut.. ☺, *terimakasih...*
18. Rekan-rekan Masjid. *Bagai sebuah lokomotif, maju tanpa kenal henti..!!*
19. Rekan-rekan WE&T. *Student today, Entrepreneur tomorrow.. You can if you
think you can.. We are big family. Salam Wetters..!! Salam Entrepreneur..!!*
20. Rekan-rekan di BEM, LM dan MKM.. *Sepurane, aku dhisikan yoo..* ☺ *Nitip
kampus ITS.. Lanjutkan perjuangan..!! Hidup Mahasiswa..!! Hidup Rakyat..!!*
21. Rekan-rekan HMTC. *Hmm.. HMTC...* ☺ *vivat TC..!!*
22. Arek-arek kost. *Aku jarang mulih..* ☺ *Kost itu tempat mandi dan cuci thok..* ☺
23. Rekan-rekan Xpertize Global Pratama. *We are the real leader company!!*
24. Semua yang telah menorehkan tinta emasnya dalam lembar hidup penulis,
Khan ku kenang dikau semua dalam lantunan syair hidupku..

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR.....	ii
UCAPAN TERIMA KASIH	iii
DAFTAR ISI	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG.....	1
1.2. TUJUAN DAN MANFAAT.....	2
1.3. PERUMUSAN MASALAH.....	2
1.4. BATASAN PERMASALAHAN	3
1.5. METODOLOGI.....	3
1.6. SISTEMATIKA PENULISAN.....	5
BAB II KONSEP SISTEM PENGUKURAN KINERJA, KONSEP QMPMS DAN METODE AHP.....	8
2.1. KONSEP SISTEM PENGUKURAN KINERJA	8
2.1.1. Definisi Sistem Pengukuran Kinerja	8
2.1.2. Mengapa Pengukuran Kinerja Diperlukan.....	10
2.1.3. Perkembangan Sistem Pengukuran Kinerja	10
2.1.4. Bagaimana Melakukan Pengukuran Kinerja.....	13
2.2. QUANTITATIVE MODEL FOR PERFORMANCE MEASUREMENT SYSTEM (QMPMS)	15
2.2.1. Deskripsi QMPMS.....	15
2.2.2. Keuntungan Penggunaan QMPMS.....	15
2.2.3. Langkah-Langkah dalam Penggunaan QMPMS	16
2.3. METODE ANALYTICAL HIERARCHY PROCESS.....	21
2.3.1. Deskripsi AHP.....	21
2.3.2. Manfaat AHP.....	21
2.3.3. Kelemahan AHP	22
2.3.4. Prosedur AHP	23
2.4. BALANCED SCORECARD.....	27



BAB III SISTEM PENGUKURAN KINERJA

DI PT MERPATI NUSANTARA AIRLINES SURABAYA	34
3.1. IDENTIFIKASI FAKTOR-FAKTOR PENGUKURAN KINERJA	34
3.3.1. Identifikasi Visi, Misi, Tujuan dan Strategi Perusahaan.....	34
3.3.2. Identifikasi Objektif MMF	36
3.3.3. Identifikasi Key Performance Indicator (KPI)	39
3.2. STRUKTUR HIRARKI ANALITIS FAKTOR-FAKTOR PENGUKURAN KINERJA DAN PELEVELANNYA	40
3.3. PEMBOBOTAN	43
3.3.1. Pembobotan Perspektif	43
3.3.2. Pembobotan pada Kriteria.....	44
3.3.3. Pembobotan pada Indikator.....	45

BAB IV DESAIN PERANGKAT LUNAK

4.1. DESAIN DATA.....	49
4.2. DESAIN PROSES	56
4.2.1. Analisis Usecase View.....	56
4.2.2. Desain Logical View	60
4.2.3. Desain Class <i>Package</i>	79
4.3. DESAIN ANTARMUKA.....	85

BAB V IMPLEMENTASI PERANGKAT LUNAK.....

5.1. IMPLEMENTASI DATA.....	87
5.2. IMPLEMENTASI PROSES	91
5.2.1. Implementasi Proses Login	91
5.2.2. Implementasi Proses Manage Departemen	91
5.2.3. Implementasi Proses Setting Periode.....	94
5.2.4. Implementasi Proses Input dan Update KPI	95
5.2.5. Implementasi Proses pada Input dan Update Bobot KPI.....	95
5.3. IMPLEMENTASI ANTARMUKA	96
5.3.1. Implementasi Antarmuka Login.....	96
5.3.2. Implementasi Antarmuka Manage Departemen	97
5.3.3. Implementasi Antarmuka Setting Periode	99
5.3.4. Implementasi Antarmuka Input dan Update KPI	100
5.3.5. Implementasi Antarmuka Input dan Update Bobot KPI.....	103

BAB VI UJI COBA DAN EVALUASI PERANGKAT LUNAK.....

6.1. LINGKUNGAN UJI COBA	105
6.1.1. Server	105

6.1.2. Client.....	106
6.2. DATA UJI COBA	106
6.3. PELAKSANAAN UJI COBA	108
6.2.1. Skenario Pertama Sebagai Aktor Administrator	109
6.2.2. Skenario Kedua Sebagai HRD	112
6.2.3. Skenario Ketiga Sebagai HLM.....	116
6.2.4. Skenario Keempat Sebagai MLLM	117
 BAB VII SIMPULAN	121
 DAFTAR PUSTAKA	122
 LAMPIRAN A.....	124

DAFTAR GAMBAR

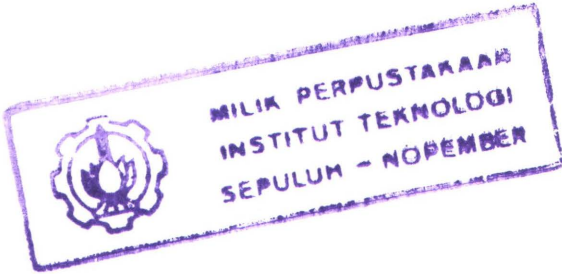
Gambar 2.1 Langkah-langkah dalam QMPMS	16
Gambar 2.2 Gambar Peta Kognitif	18
Gambar 2.3 Diagram Sebab dan Akibat.....	19
Gambar 2.4 Diagram Pohon	20
Gambar 2.5 Diagram Hirarki AHP	23
Gambar 2.6 Framework dalam Balanced Scorecard untuk menerjemahkan strategi ke dalam proses-proses operasional.....	29
Gambar 2.7 Perspektif Konsumen-Core Measures.....	31
Gambar 2.8 Network Pengukuran Perspektif Tumbuh dan Belajar	33
Gambar 3.1 Proses Bisnis MMF	36
Gambar 3.2 Struktur Hirarki KPI.....	42
Gambar 4.1 Data Model Diagram.....	50
Gambar 4.2 Aktor-Aktor yang terlibat dalam sistem.....	56
Gambar 4.3 UseCase-UseCase dalam sistem	59
Gambar 4.4 Usecase Realization Administrator.....	61
Gambar 4.5 Usecase Realization HRD	62
Gambar 4.6 Usecase Realization HLM.....	63
Gambar 4.7 Usecase Realization MLLM.....	64
Gambar 4.8 Activity Diagram Input dan Update KPI.....	66
Gambar 4.9 Sequence Diagram Input dan Update KPI	66
Gambar 4.10 Activity Diagram Setting Periode.....	67
Gambar 4.11 Sequence Diagram Setting Periode.....	68
Gambar 4.12 Activity Diagram Input dan Update Bobot KPI	69
Gambar 4.13 Sequence Diagram Input dan Update Bobot KPI	69
Gambar 4.14 Activity Diagram Setting KPI Aktif Periodik	70
Gambar 4.15 Sequence Diagram Setting KPI Aktif Periodik	71
Gambar 4.16 Activity Diagram AHP.....	72
Gambar 4.17 Sequence Diagram AHP.....	72
Gambar 4.18 Activity Diagram Menentukan Target KPI Periodik	73
Gambar 4.19 Sequence Diagram Menentukan Target KPI Periodik	74
Gambar 4.20 Activity Diagram Melaporkan Pencapaian KPI Periodik	75
Gambar 4.21 Sequence Diagram Melaporkan Pencapaian KPI Periodik	75
Gambar 4.22 Activity Diagram Accountability.....	76
Gambar 4.23 Sequence Diagram Accountability	77
Gambar 4.24 Activity Diagram Lihat Hasil Pengukuran Kinerja	78
Gambar 4.25 Sequence Diagram Lihat Hasil Pengukuran Kinerja	78
Gambar 4.26 Package JavaBeans	79
Gambar 4.27 Class-class pada Package Connection.....	80
Gambar 4.28 Class-class pada Package Controller.....	82
Gambar 4.29 Class-class pada Package Html	83
Gambar 4.30 Class-class pada Package Model	84
Gambar 4.31 Class-class pada Package Util	85
Gambar 4.32 Desain Antarmuka	86

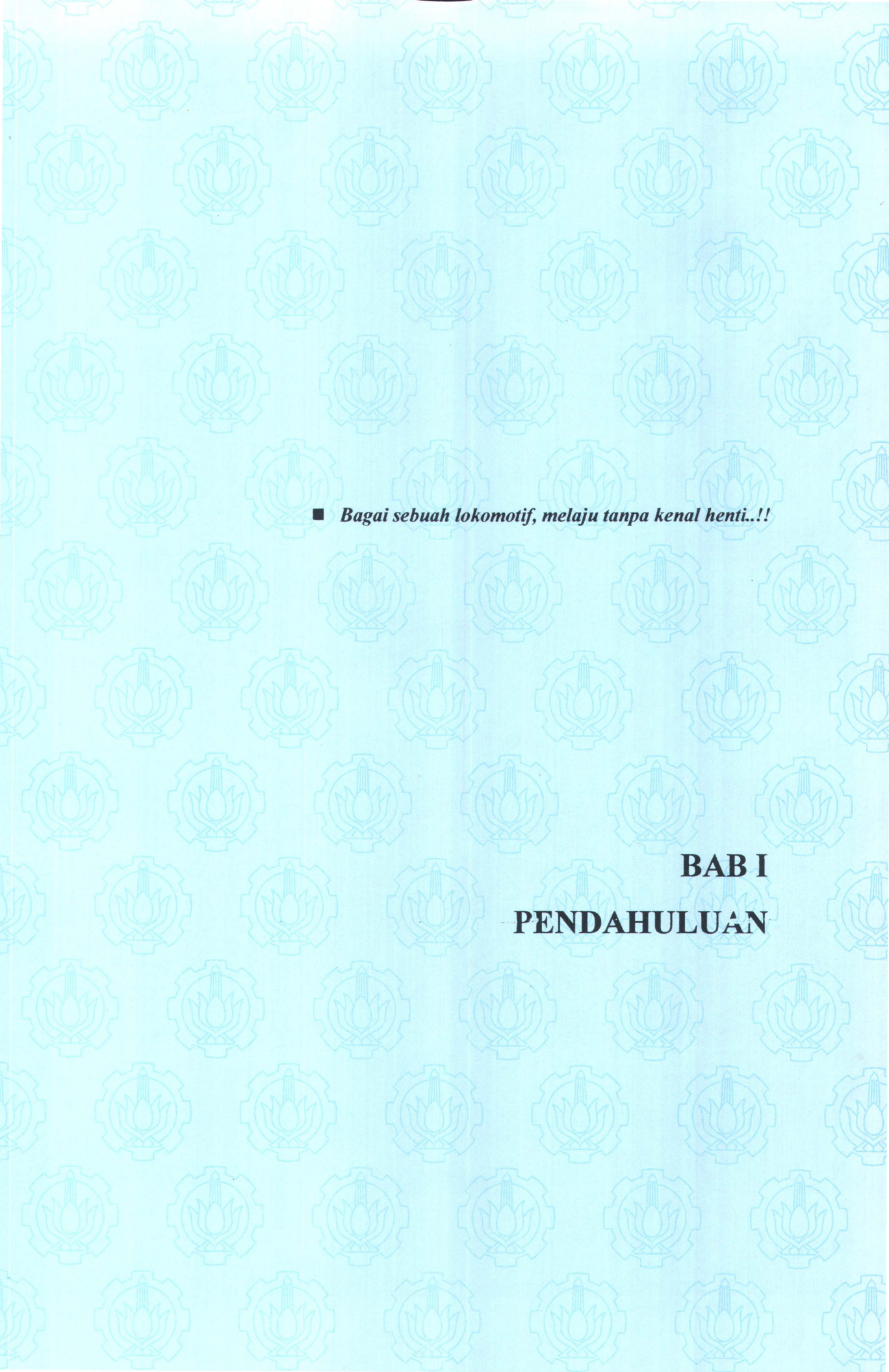
Gambar 5.1 Implementasi Tabel TUser	87
Gambar 5.2 Implementasi Tabel TDept	88
Gambar 5.3 Implementasi Tabel TPeriode.....	88
Gambar 5.4 Implementasi Tabel TKPI	88
Gambar 5.5 Implementasi Tabel TWeight.....	89
Gambar 5.6 Implementasi Tabel TPerformance.....	89
Gambar 5.7 Implementasi Tabel TAcc	89
Gambar 5.8 Implementasi Tabel TRole	90
Gambar 5.9 Implementasi Tabel TMenu	90
Gambar 5.10 Implementasi Tabel TUserRoleMenu	90
Gambar 5.11 Penggalan implementasi loginBean.java.....	92
Gambar 5.12 Penggalan implementasi roleBean.java.....	93
Gambar 5.13 Penggalan implementasi dept_edit_act.jsp.....	94
Gambar 5.14 Penggalan implementasi dept_tambah_act.jsp	94
Gambar 5.15 Penggalan implementasi periode_act.jsp	95
Gambar 5.16 Implementasi Antarmuka Login	97
Gambar 5.17 Tampilan Antarmuka Login	97
Gambar 5.18 Implementasi Antarmuka Manage Departemen	98
Gambar 5.19 Implementasi Antarmuka Tambah Departemen	98
Gambar 5.20 Tampilan Antarmuka Manage Departemen	99
Gambar 5.21 Implementasi Antarmuka Setting Periode.....	99
Gambar 5.22 Tampilan Antarmuka Seting Periode	100
Gambar 5.23 Implementasi Antarmuka tree_edit_right.jsp	100
Gambar 5.24 Implementasi Antarmuka tree_edit_node_info.jsp	101
Gambar 5.25 Implementasi Antarmuka tree_edit_left.jsp	102
Gambar 5.26 Tampilan Antarmuka Input dan Update KPI.....	102
Gambar 5.27 Implementasi Antarmuka tree_weight_bottom.jsp.....	103
Gambar 5.28 Tampilan Antarmuka Input dan Update Bobot KPI	103
Gambar 5.29 Implementasi Antarmuka tree_weight_top.jsp.....	104
Gambar 6.1 Login Pengguna admin sebagai Aktor Administrator	109
Gambar 6.2 Halaman Utama Aplikasi	110
Gambar 6.3 Manage Pengguna dan Role	110
Gambar 6.4 Edit Data Pengguna dan Password	111
Gambar 6.5 Manage Departemen	111
Gambar 6.6 Tambah Departemen Baru.....	112
Gambar 6.7 Halaman Utama Login hrd sebagai aktor HRD.....	113
Gambar 6.8 Setting Periode.....	113
Gambar 6.9 Konfirmasi Menghapus Periode	114
Gambar 6.10 Input dan Update KPI.....	114
Gambar 6.11 Input dan Update Bobot KPI dengan metode AHP	115
Gambar 6.12 Halaman Utama HLM.....	116
Gambar 6.13 Menentukan Target KPI Periodik	117
Gambar 6.14 Melaporkan pencapaian KPI periodik.....	118
Gambar 6.15 Melakukan Pengukuran Kinerja	119

DAFTAR TABEL

<i>Tabel 2.1 Tabel Matriks Perbandingan Berpasangan</i>	2-
<i>Tabel 2.2 Intensitas Kepentingan</i>	25
<i>Tabel 3.1 Daftar Narasumber Wawancara Identifikasi Objektif</i>	37
<i>Tabel 3.2 Hasil Penentuan Objektif Awal</i>	37
<i>Tabel 3.3 Penyesuaian Objektif Final</i>	38
<i>Tabel 3.4 Hasil Identifikasi KPI</i>	40
<i>Tabel 3.5 Hasil Perhitungan Nilai Bobot Perspektif</i>	43
<i>Tabel 3.6 Hasil Perhitungan Nilai Bobot Perspektif Finansial</i>	44
<i>Tabel 3.7 Hasil Perhitungan Nilai Bobot Perspektif Konsumen</i>	44
<i>Tabel 3.8 Hasil Perhitungan Nilai Bobot Perspektif Proses Bisnis Internal</i>	44
<i>Tabel 3.9 Hasil Perhitungan Nilai Bobot Perspektif Tumbuh dan Belajar</i>	44
<i>Tabel 3.10 Hasil Perhitungan Nilai Bobot Kriteria Profit</i>	45
<i>Tabel 3.11 Hasil Perhitungan Nilai Bobot Kriteria Akuntabilitas</i>	45
<i>Tabel 3.12 Hasil Perhitungan Nilai Bobot Kriteria Kepuasan Pelanggan</i>	45
<i>Tabel 3.13 Hasil Perhitungan Nilai Bobot Meningkatkan Market Share</i>	46
<i>Tabel 3.14 Hasil Perhitungan Nilai Bobot Kriteria Kualitas</i>	46
<i>Tabel 3.15 Hasil Perhitungan Nilai Bobot Kriteria Produktivitas</i>	46
<i>Tabel 3.16 Hasil Perhitungan Nilai Bobot Kriteria SDM</i>	47
<i>Tabel 3.17 Hasil Perhitungan Nilai Bobot Kriteria Sistem Informasi</i>	47
<i>Tabel 3.18. Hasil Perhitungan Nilai Bobot Seluruh KPI</i>	47
<i>Tabel 4.1 Penjelasan Tabel TUser</i>	51
<i>Tabel 4.2 Penjelasan Tabel TDept</i>	51
<i>Tabel 4.3 Penjelasan Tabel TPeriode</i>	52
<i>Tabel 4.4 Penjelasan Tabel TKPI</i>	52
<i>Tabel 4.5 Penjelasan Tabel TWeight</i>	53
<i>Tabel 4.6 Penjelasan Tabel TPerformance</i>	54
<i>Tabel 4.7 Penjelasan Tabel TAcc</i>	54
<i>Tabel 4.8 Penjelasan Tabel TRole</i>	55
<i>Tabel 4.9 Penjelasan Tabel TMenu</i>	55
<i>Tabel 4.10 Penjelasan Tabel TMenu</i>	55
<i>Tabel 4.11 Aktor dan definisi tugasnya dalam sistem</i>	57
<i>Tabel 4.12 Daftar Usecase</i>	58
<i>Tabel 6.1 Data Ujicoba Pengguna dan Password</i>	106
<i>Tabel 6.2 Data Ujicoba Periode</i>	106
<i>Tabel 6.3 Data Ujicoba Target</i>	107
<i>Tabel 6.4 Data Ujicoba Pencapaian</i>	107
<i>Tabel A.1 Nilai Intensitas Kepentingan</i>	124
<i>Tabel A.2 Contoh Tabel Isian Matriks</i>	124
<i>Tabel A.3 Contoh Hasil Pengisian Matriks</i>	125
<i>Tabel A.4 Tabel Isian Overall</i>	125
<i>Tabel A.5 Tabel Isian Perspektif Konsumen</i>	125
<i>Tabel A.6 Tabel Isian Kriteria Kepuasan Pelanggan</i>	126
<i>Tabel A.7 Tabel Isian Kriteria Market Share</i>	126

Tabel A.8 Tabel Isian Perspektif Finansial 126
Tabel A.9 Tabel Isian Kriteria Profit 127
Tabel A.10 Tabel Isian Kriteria Meningkatkan Akuntabilitas Keuangan..... 127
Tabel A.11 Tabel Isian Perspektif Proses Bisnis Internal..... 127
Tabel A.12 Tabel Isian Kriteria Kualitas 127
Tabel A.13 Tabel Isian Kriteria Produktivitas 128
Tabel A.14 Tabel Isian Perspektif Proses tumbuh dan Belajar 128
Tabel A.15 Tabel Isian Kriteria SDM 129
Tabel A.16 Tabel Isian Kriteria Sistem Informasi 129





■ *Bagai sebuah lokomotif, melaju tanpa kenal henti..!!*

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Pada tingkat persaingan global yang kompetitif, perusahaan saling berlomba dalam memberikan produk dan pelayanan yang terbaik bagi pelanggannya. Dalam memenangkan persaingan global tersebut, perusahaan sebagai organisasi bisnis, memiliki visi, misi dan tujuan yang pada implementasinya dijabarkan dalam sejumlah sasaran perusahaan. Dalam rangka menuju visi, misi dan tujuan tersebut, perusahaan memiliki strategi jangka panjang dan juga strategi jangka pendek untuk mencapai sasaran-sasaran yang dimaksud.

Adanya ketidakpastian lingkungan dalam pengimplementasian strategi yang ada dapat mengakibatkan terjadinya penyimpangan pencapaian strategi atau bahkan dapat menimbulkan adanya strategi yang tidak terealisasi. Untuk mengetahui perkembangan pencapaian sasaran perusahaan, diperlukan suatu ukuran yang menggambarkan perkembangan kinerja yang ada.

Dalam melakukan pengukuran, perlu diketahui indikator-indikator kinerja kunci (*Key Performance Indicators/KPI*) yang menggambarkan *milestone* pencapaian sasaran jangka pendek dan jangka panjang. Dari setiap gambaran tahapan yang didapat, dalam hal ini pencapaian kinerja, dapat diformulasikan *corrective action* atas terjadinya penyimpangan pencapaian sasaran atau strategi yang tidak terealisasi untuk dapat dilakukan revisi.

PT Merpati Nusantara Airlines Surabaya, sebagai salah satu perusahaan transportasi udara terkemuka di Indonesia, sangat berkepentingan untuk dapat bertahan dan memenangkan persaingan bisnis transportasi udara di Indonesia. Oleh karena itu, sebuah perangkat lunak pengukuran kinerja strategis yang handal, fleksibel, dan mudah diakses, sangat dibutuhkan oleh pihak manajemen PT Merpati Nusantara Airlines Surabaya agar supaya pelaksanaan pengukuran kinerja dapat dilakukan dengan lebih mudah, cepat dan akurat.

1.2. TUJUAN DAN MANFAAT

Tujuan dari tugas akhir ini adalah untuk mendesain dan mengimplementasikan sebuah perangkat lunak pengukuran kinerja yang berbasis web dengan menggunakan metode *Analytical Hierarchy Process* (AHP). Dengan tersedianya sistem dan perangkat lunak pengukuran kinerja, maka PT Merpati Nusantara Airlines Surabaya dapat melakukan pengukuran kinerja strategis yang handal, fleksibel, dan dapat diakses dengan mudah.

1.3. PERUMUSAN MASALAH

Untuk memenuhi tujuan yang diinginkan, perlu dirumuskan beberapa pokok permasalahan berikut:

- a. Bagaimana melakukan pembobotan dengan menggunakan metode AHP (*Analytical Hierarchy Process*).
- b. Bagaimana melakukan analisis, desain dan implementasi perangkat lunak berbasis web dalam melakukan pengukuran kinerja manajemen strategis perusahaan.

1.4. BATASAN PERMASALAHAN

Permasalahan yang dibahas dan diselesaikan dalam Tugas Akhir ini meliputi:

- a. Faktor-faktor pengukuran KPI didasarkan pada data yang diperoleh dari Direktorat Teknik PT Merpati Nusantara Airlines Surabaya dalam periode waktu pengerjaan Tugas Akhir.
- b. Analisis proses bisnis dibatasi pada Direktorat Teknik PT Merpati Nusantara Airlines Surabaya yang terkait dengan sistem pengukuran kinerja.
- c. Analisis dan Desain perangkat lunak dilakukan dengan menggunakan metode berorientasi obyek.

1.5. METODOLOGI

Metodologi yang digunakan dalam pembuatan Tugas Akhir ini adalah sebagai berikut:

a. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk memahami lebih mendalam tentang metode AHP (*Analytical Hierarchy Process*). Termasuk diantaranya adalah pencarian literatur, diskusi dan pemahaman terhadap topik Tugas Akhir.

Pada tahap ini juga akan dipelajari metodologi dan algoritma yang akan digunakan dalam desain perangkat lunak berorientasi obyek dan implementasi perangkat lunak yang berbasis web.

b. Identifikasi dan Analisis KPI

Identifikasi dan analisis faktor-faktor pengukuran KPI dilakukan dengan menggunakan metode AHP pada Direktorat Teknik PT Merpati Nusantara Airlines Surabaya.

c. Desain perangkat lunak

Tahap ini adalah tahap desain perangkat lunak yang akan dibuat. Dalam tahap desain perangkat lunak ini, penulis akan menggunakan desain perangkat lunak yang berorientasi obyek dan Rational Rose Enterprise Edition sebagai *tools*.

d. Pembuatan perangkat lunak

Dalam tahap ini, dilakukan implementasi berdasarkan desain perangkat lunak yang telah dibuat pada tahap sebelumnya.

e. Uji coba dan Evaluasi

Pada tahap ini dilakukan uji coba program untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan jika ada kekurangan.

Ujicoba perangkat lunak meliputi uji coba fungsional dan teknis perangkat lunak pada Direktorat Teknik PT Merpati Nusantara Airlines Surabaya.

f. Penyusunan Buku Tugas Akhir

Pada tahap terakhir ini disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir. Dokumentasi ini dibuat agar memudahkan orang lain yang ingin mengembangkan perangkat lunak sistem informasi ini lebih lanjut.

1.6. SISTEMATIKA PENULISAN

Sistematika penulisan dalam Tugas Akhir ini di bagi dalam delapan bab, dengan rincian sebagai berikut:

- **BAB I : PENDAHULUAN**

Bagian ini berisi penjelasan mengenai latar belakang, tujuan dan manfaat, perumusan masalah, batasan permasalahan, metodologi serta sistematika penulisan Tugas Akhir.

- **BAB II : KONSEP SISTEM PENGUKURAN KINERJA, AHP DAN QMPMS**

Bab ini menjelaskan mengenai konsep sistem pengukuran kinerja, konsep *Quantitative Model for Performance Measurement System* (QMPMS) sebagai model sistem pengukuran kinerja yang dipakai dalam Tugas Akhir, dan *Analytical Hierarchy Process* (AHP) sebagai metode untuk pembobotan KPI.

- **BAB III : SISTEM PENGUKURAN KINERJA DI PT MERPATI NUSANTARA AIRLINES SURABAYA**

Bab ini akan menjelaskan mengenai sistem pengukuran kinerja yang digunakan di PT Merpati Nusantara Airlines Surabaya, yang terdiri dari identifikasi faktor-faktor pengukuran kinerja dan pembobotannya, dan struktur hirarki analitis faktor-faktor pengukuran kinerja.

- **BAB IV : DESAIN PERANGKAT LUNAK**

Bab ini menjelaskan tentang tahapan proses desain perangkat lunak. Desain dilakukan dengan menggunakan konsep *Visual Modeling*, yaitu konsep atau cara berpikir tentang suatu permasalahan dengan menggunakan pemodelan. Model berguna untuk memahami kebutuhan, komunikasi, dokumentasi dan desain program serta database. Selain itu, desain juga dilakukan dengan menggunakan metode berorientasi objek.

- **BAB V : IMPLEMENTASI PERANGKAT LUNAK**

Bab ini menjelaskan tentang tahap implementasi perangkat lunak. Tahap implementasi perangkat lunak dilakukan setelah tahap desain selesai. Meskipun pada pelaksanaannya, akan ada proses dimana desain yang telah dibuat perlu diubah kembali untuk menyesuaikan dengan proses penulisan program. Class-class yang telah didefinisikan pada tahap desain akan diimplementasikan pada bahasa pemrograman Java. Dan untuk desain *usecase-usecase* pada bagian view akan diimplementasikan dalam bahasa JavaServer Pages.

- **BAB VI : UJICoba DAN EVALUASI PERANGKAT LUNAK**

Bab ini menjelaskan tentang tahap ujicoba dan evaluasi perangkat lunak. Pengujian pada perangkat lunak / aplikasi sistem pengukuran kinerja dapat dilakukan setelah proses perancangan dan implementasi selesai. Uji coba dilakukan untuk mengetahui apakah aplikasi yang telah dibuat dapat memecahkan permasalahan-permasalahan yang menjadi latar belakang dibuatnya aplikasi tersebut.

- **BAB VII : KESIMPULAN DAN SARAN**

Bab ini menjelaskan tentang kesimpulan dan saran Tugas Akhir. Aplikasi pengukuran kinerja yang dihasilkan dapat berjalan dengan merealisasikan tujuan dari pembuatannya yaitu mampu melakukan pengukuran kinerja manajemen strategis perusahaan.

Student today, Entrepreneur tomorrow..

You can if you think you can..

We are big family. Salam Wetters..!!

Salam Entrepreneur..!!

BAB II

KONSEP SISTEM PENGUKURAN KINERJA, KONSEP QMPMS DAN METODE AHP



BAB II

KONSEP SISTEM PENGUKURAN KINERJA, KONSEP QMPMS DAN METODE AHP

Bab ini menjelaskan mengenai konsep sistem pengukuran kinerja, konsep *Quantitative Model for Performance Measurement System* (QMPMS) sebagai model sistem pengukuran kinerja yang dipakai dalam Tugas Akhir, dan *Analytical Hierarchy Process* (AHP) sebagai metode untuk pembobotan KPI.

2.1. KONSEP SISTEM PENGUKURAN KINERJA

Subbab ini menjelaskan mengenai konsep sistem pengukuran kinerja, yang terdiri dari definisi sistem pengukuran kinerja, mengapa perlu dilakukan pengukuran kinerja, perkembangan sistem pengukuran kinerja, dan bagaimana melakukan pengukuran kinerja.

2.1.1. Definisi Sistem Pengukuran Kinerja

Menurut U.S. General Accounting, pengukuran kinerja dapat didefinisikan seperti berikut [ART01]:

"suatu aktivitas pemantauan dan pelaporan secara terus menerus terhadap pencapaian program terutama kemajuan ke arah pencapaian sasaran. Di dalam pengukuran kinerja disebutkan tentang tingkat dan aktivitas yang berhubungan dengan program, keluaran dari program baik berupa produk secara langsung maupun jasa, serta hasil dari produk atau jasa tersebut.

Program yang dimaksud tersebut dapat berupa aktivitas, proyek, fungsi atau kebijakan yang mengidentifikasi tujuan”.

Pengukuran kinerja secara kuantitatif akan dapat memberikan informasi kepada perusahaan mengenai produk, jasa serta proses yang berlangsung dalam perusahaan tersebut. Pengukuran kinerja adalah sebuah fasilitas (*tool*) yang dapat membantu pihak manajemen untuk memahami, mengelola dan memperbaiki apa yang dilakukan oleh suatu organisasi.

Suatu pengukuran kinerja yang efektif harus mengandung informasi-informasi sebagai berikut [SUW03]:

- Seberapa baik hasil dari apa yang dilakukan suatu organisasi.
- Apakah laju perkembangan suatu organisasi telah sesuai ke arah pencapaian sasarannya.
- Seberapa besar kepuasan pelanggan.
- Apakah proses telah berada dalam pemantauan berdasarkan pada data statistik (*under statistical control*).
- Apakah perlu adanya suatu perbaikan dalam organisasi.

Informasi-informasi diatas akan sangat membantu pihak manajemen dalam proses perumusan keputusan penting terhadap keberlangsungan perusahaan. Pengukuran kinerja tersusun atas nomor yang menyatakan jumlah dan satuan pengukuran yang menyatakan arti dari jumlah tersebut. Pengukuran kinerja akan selalu terkait dengan sasaran dan tujuan, yang merupakan target perusahaan.

2.1.2. Mengapa Pengukuran Kinerja Diperlukan

Beberapa organisasi terkemuka dan beberapa pakar pengukuran kinerja memaparkan beberapa alasan berikut mengenai mengapa pengukuran kinerja diperlukan [ART01]:

- Pengukuran kinerja sangat bermanfaat bagi organisasi, karena menyediakan suatu pendekatan terstruktur yang terfokus pada rencana strategis, sasaran, dan kinerja.
- Pengukuran lebih memfokuskan suatu organisasi pada apa yang akan diselesaikan dan memaksa untuk lebih berkonsentrasi pada waktu, sumber daya, dan energi dalam mencapai tujuan yang diinginkan.
- Pengukuran kinerja dapat memperbaiki komunikasi internal karyawan dan eksternal organisasi dengan para pelanggan maupun *stakeholder*.
- Pengukuran kinerja dapat membantu suatu organisasi untuk mempertanggungjawabkan program serta biaya yang diperlukan.

2.1.3. Perkembangan Sistem Pengukuran Kinerja

Perkembangan sistem pengukuran kinerja yang benar-benar tepat baru dimulai setelah adanya sistem industri pada saat revolusi industri di awal abad ke-19. Perkembangan sistem pengukuran kinerja dikelompokkan menjadi beberapa periode, yaitu [SUW01]:

a. Sistem Pengukuran Kinerja untuk Mengukur Efisiensi Proses Bisnis Internal (1800 – 1900)

Pada periode ini, perusahaan membuat matrik-matrik untuk mengukur efisiensi dari proses aktivitas tunggal, seperti *cost/lb*, *cost/hour*, *standard labour cost* dan *standard material cost*. Matrik-matrik tersebut digunakan untuk mengukur efisiensi aktivitas manufaktur. *Cost/ton-mile* digunakan untuk mengukur efisiensi dari aktivitas transportasi, sedangkan *inventory stock-turn* digunakan untuk mengukur efisiensi aktivitas logistik dan distribusi.

b. Sistem Pengukuran Kinerja untuk Mengukur Profitabilitas Unit dan Organisasi secara Keseluruhan (1900 – 1925)

Pembentukan perusahaan yang terintegrasi secara vertikal (*vertically integrated firms*) pada masa itu didorong oleh keberhasilan dalam menggabungkan beberapa *single stage* dan *domestic process* menjadi suatu perusahaan dengan aktivitas tunggal. Beberapa aktivitas seperti *manufacturing*, *purchasing*, *transportation* dan *distribution* yang sebelumnya merupakan aktivitas yang terisolasi dari suatu perusahaan yang independen, sekarang menjadi aktivitas-aktivitas terintegrasi dari perusahaan dengan berbagai aktivitas (*multi activity organization*). Hal tersebut menyebabkan diperlukannya ukuran kinerja baru yang dapat digunakan untuk mengintegrasikan berbagai ukuran kinerja. Pada akhirnya, Du Pont menemukan *Return Of Investment* (ROI) yang dapat digunakan untuk mengukur kinerja unit-unit organisasi dan kinerja perusahaan secara

menyeluruh. Selain itu, teknik untuk menganalisis *cost variance* juga ditemukan pada periode ini.

c. Relevance Lost (1925 – 1980)

Sampai tahun 1925, seluruh ukuran kinerja tradisional untuk mengukur biaya produksi dan profitabilitas perusahaan seperti biaya pembuatan produk dan variasinya, ROI, profitabilitas dan rasio-rasio keuangan telah ditemukan. Meskipun terjadi perubahan-perubahan pada lingkungan bisnis, namun belum ada inovasi yang fundamental pada periode ini. Hal tersebut menyebabkan sistem pengukuran kinerja tidak lagi relevan dengan lingkungan bisnis.

d. Perbaikan Sistem Akuntansi Biaya dan Pembuatan Ukuran Kinerja Non-Finansial secara Individual (1980 – 1990's)

Setelah tumbuh kesadaran bahwasanya akuntansi biaya tradisional tidak mampu lagi menyediakan informasi yang cepat, akurat, dan relevan yang dapat digunakan oleh manajer untuk mengontrol operasi perusahaan dan menstimulasi dengan tepat perilaku karyawan, orang-orang mulai mencari sistem akuntansi biaya dan sistem pengukuran kinerja baru. *Activity based Costing* ditemukan untuk dapat menghitung penyerapan biaya yang lebih akurat.

Setelah itu, manajer dan peneliti menyadari bahwa finansial bukanlah merupakan satu-satunya ukuran kinerja yang menjamin kelangsungan hidup perusahaan dalam jangka panjang. Banyak ukuran kinerja lain, seperti kualitas, kecepatan pengiriman, yang bukan dari indikator finansial mulai

digunakan. Sehingga, disamping ukuran kinerja finansial, manajer mulai menggunakan juga ukuran kinerja non finansial. Tetapi kedua ukuran kinerja tersebut masih digunakan secara sendiri-sendiri dan belum terintegrasi.

e. Sistem Pengukuran Kinerja Terintegrasi (1990 – Sekarang)

Para manajer dan peneliti menyadari bahwa indikator kinerja finansial dan non finansial harus dibentuk ke dalam satu bagian sistem pengukuran kinerja yang terintegrasi dan koheren, dimana kedua indikator tersebut harus dipergunakan secara seimbang. Akibatnya, kemudian berkembang penelitian-penelitian untuk membangun sistem pengukuran kinerja yang terintegrasi.

2.1.4. Bagaimana Melakukan Pengukuran Kinerja

Dalam melakukan pengukuran kinerja terdapat beberapa macam cara yang berbeda. Namun, secara umum terdapat hal-hal yang pada keadaan standar selalu dilakukan pada semua jenis pengukuran kinerja, antara lain:

a. Identifikasi faktor-faktor pengukuran kinerja

Hal ini dilakukan pada awal aktifitas pengukuran kinerja, karena faktor-faktor pengukuran kinerja tersebut selanjutnya akan digunakan sebagai parameter dalam mengukur kinerja. Umumnya, identifikasi ini diperoleh berdasarkan visi, misi dan tujuan perusahaan terkait, serta berdasarkan proses bisnis yang berjalan pada suatu perusahaan.

b. Pembobotan terhadap setiap faktor pengukuran kinerja

Pembobotan merupakan nilai kuantitatif besar kecilnya pengaruh setiap faktor pengukuran terhadap kinerja perusahaan. Untuk menghitung

bobot setiap faktor terhadap pengukuran kinerja dapat digunakan beberapa metode, salah satunya adalah dengan metode *Analytical Hierarchy Process* (AHP).

c. Menentukan setiap jangka waktu periode pengukuran kinerja

Jangka waktu periode merupakan lamanya setiap periode pengukuran kinerja. Misalnya saja periode pengukuran kinerja selama tiga bulanan atau tiap semester. Semakin pendek jangka waktu periode pengukuran kinerja maka akan semakin rinci pula perkembangan kinerja perusahaan dapat dipantau.

d. Menentukan target yang harus dicapai setiap faktor pengukuran kinerja

Pada setiap awal periode pengukuran kinerja, pihak manajemen menentukan target yang harus dicapai oleh perusahaan terhadap setiap faktor pengukuran kinerja. Target inilah yang nantinya menjadi acuan mengenai seberapa tinggi suatu kinerja harus dicapai oleh perusahaan.

e. Mencatat realisasi pencapaian setiap faktor pengukuran kinerja

Pada setiap akhir periode pengukuran kinerja, dilakukan pencatatan realisasi pencapaian setiap faktor pengukuran kinerja. Selanjutnya pencapaian realisasi tersebut dibandingkan dengan target, dan dapat diketahui berapa tingkat kinerja perusahaan terhadap target yang telah ditetapkan pada awal periode.

2.2. QUANTITATIVE MODEL FOR PERFORMANCE MEASUREMENT SYSTEM (QMPMS)

Pada subbab ini dijelaskan mengenai *Quantitative Model for Performance Measurement System* (QMPMS) sebagai model konsep pengukuran kinerja, yang terdiri dari deskripsi QMPMS, keuntungan menggunakan QMPMS, dan langkah-langkah dalam penggunaan QMPMS.

2.2.1. Deskripsi QMPMS

QMPMS yang merupakan kependekan dari *Quantitative Model for Performance Measurement System* merupakan fasilitas (*tool*) dalam memudahkan pembobotan dalam perancangan pengukuran kinerja [ARI04]. QMPMS dikembangkan untuk memudahkan dalam :

- a. Identifikasi dari faktor-faktor yang berpengaruh terhadap kinerja dan hubungannya.
- b. Pembuatan struktur hirarki faktor-faktor yang berpengaruh terhadap kinerja.
- c. Penghitungan hubungan antara faktor satu dengan yang lainnya dan pada kinerja secara keseluruhan.

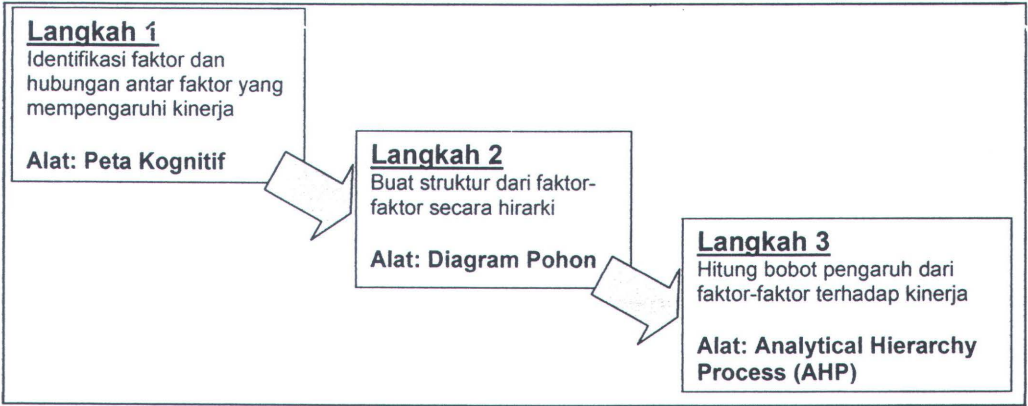
2.2.2. Keuntungan Penggunaan QMPMS

Dengan penggunaan model QMPMS memiliki banyak keuntungan. Keuntungan-keuntungan yang dapat diperoleh dengan menggunakan sistem pengukuran kinerja QMPMS adalah [ARI04]:

- a. Dapat digunakan pada organisasi profit ataupun non profit.
- b. Faktor yang mempengaruhi kinerja dapat diidentifikasi dan efeknya dapat dikuantifikasi.
- c. Efek faktor multidimensi terhadap kinerja dapat diagregasi ke dalam unit dimensi tunggal (persentase).
- d. Membantu manajer mengkuantifikasikan tingkat pengaruh tiap faktor terhadap kinerja secara keseluruhan, sehingga dapat membantu dalam memfokuskan tindakan perbaikan.
- e. Hubungan antar faktor dapat diidentifikasi dengan jelas dan diekspresikan dalam bentuk kuantitatif.
- f. Membantu dalam pemahaman dinamika faktor yang mempengaruhi kinerja.
- g. Memfasilitasi untuk reduksi jumlah laporan pengukuran kinerja.

2.2.3. Langkah-Langkah dalam Penggunaan QMPMS

Penggunaan metode QMPMS memiliki beberapa langkah-langkah yang harus dilakukan, seperti terlihat pada gambar 2.1 berikut:



Gambar 2.1 Langkah-langkah dalam QMPMS

Berikut ini dijelaskan detail mengenai langkah-langkah dalam penggunaan QMPMS tersebut.

a. Identifikasi Faktor-Faktor yang Berpengaruh terhadap Kinerja dan Hubungannya.

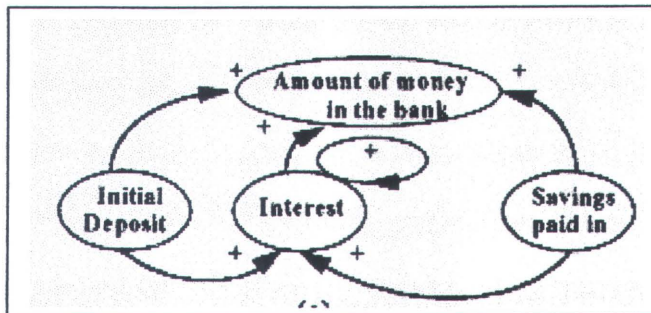
Untuk menggali dan mengidentifikasi faktor yang berpengaruh terhadap kinerja dan hubungannya, QMPMS menggunakan peta kognitif. Langkah ini merupakan tahap yang sangat penting dalam mengimplementasikan QMPMS. Kegagalan dalam memasukkan seluruh faktor yang berpengaruh terhadap kinerja dan mengidentifikasi hubungannya, akan menyebabkan keburukan pada hasil yang didapat.

Pada umumnya pengukuran kinerja melibatkan sejumlah faktor dan orang dari berbagai departemen. Faktor yang diukur seringkali tidak didefinisikan dengan baik dari tahap awal dan tidak meliputi semua komponen yang menjadi faktor nyata untuk diukur, dan penentuan hubungan antar komponen tidak selalu menjadi hal yang mudah untuk dilakukan. Individu yang berbeda akan mengartikan secara berbeda, bergantung pada pemikiran mereka sendiri.

Peta kognitif adalah alat yang efektif untuk membantu mengidentifikasi faktor-faktor yang mempengaruhi kinerja dan hubungannya. *"The Cognitive Map is a signed digraph including the way individuals, groups, and experts realise and understand a problem as well as the bilaterally connected elements (Lee et al, 1992)"* [CAL98]. Peta kognitif merupakan sebuah teknik pemodelan yang dapat digunakan untuk menggambarkan ide, keyakinan, nilai dan sikap serta hubungan satu dengan yang lainnya dalam bentuk yang dapat dipercaya, serta untuk mudah



dipelajari dan dianalisis. Gambar 2.2 berikut memperlihatkan peta kognitif dengan anak panah menunjukkan akibat dari faktor terhadap kinerja atau pada faktor lainnya.



Gambar 2.2 Gambar Peta Kognitif

Secara umum, efek dari suatu faktor terhadap kinerja seperti yang terlihat dalam gambar 2.2 dapat diklasifikasikan seperti berikut:

- **Efek langsung (vertikal)**

Efek langsung suatu faktor terhadap kinerja adalah kumpulan semua efek faktor terhadap kinerja yang melalui faktor itu. Efek langsung dari suatu faktor yang ditimbulkan oleh efek yang melekat padanya, efek interaksi sendiri, dan efek langsung faktor lain melalui faktor tersebut. Tanda positif dibagian akhir anak panah menunjukkan efek positif dari faktor

- **Efek tidak langsung (horizontal)**

Efek tidak langsung adalah efek suatu faktor terhadap kinerja melalui faktor lain dalam tingkat yang sama

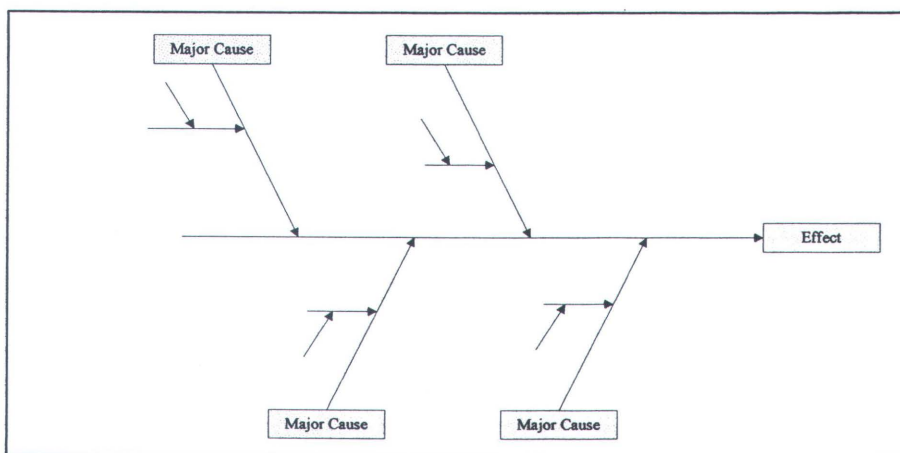
- **Efek interaksi sendiri**

Efek interaksi sendiri adalah efek suatu faktor terhadap dirinya sendiri

b. Pembuatan Struktur Hirarki Faktor

Diagram sebab dan akibat dapat digunakan untuk mendefinisikan struktur hirarki faktor. Diagram sebab dan akibat merupakan diagram yang terdiri dari garis dan simbol yang dirancang untuk menyajikan hubungan yang berarti antara suatu akibat dan penyebabnya. Diagram sebab dan akibat digunakan untuk menyelidiki hal apa yang menjadi akibat negatif dan melakukan tindakan untuk memperbaiki penyebabnya atau akibat positif dan mempelajari penyebab yang menimbulkankannya.

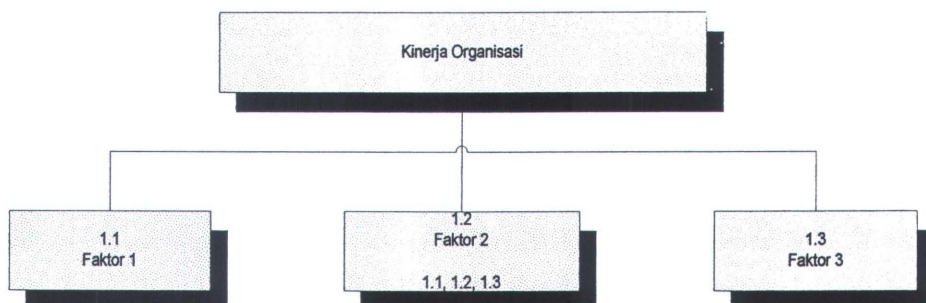
Suatu faktor menjadi anggota tingkat 0 jika faktor ini diakibatkan oleh faktor lain, tetapi tidak mempengaruhi faktor lain. Sedangkan faktor yang secara langsung berpengaruh pada faktor di tingkat tertentu ditentukan menjadi anggota tingkat bawah berikutnya. Gambar 2.3 berikut menggambarkan contoh diagram sebab dan akibat dengan akibat di sebelah kanan dan penyebab di sebelah kiri. Efeknya adalah karakteristik yang membutuhkan perbaikan. Penyebab dapat diuraikan menjadi penyebab utama seperti metode kerja, material, pengukuran, pekerja, peralatan, dan lingkungan.



Gambar 2.3 Diagram Sebab dan Akibat

Setiap penyebab utama lebih jauh lagi dibagi kedalam banyak penyebab minor, sebagai contoh, di penyebab utama metode kerja, dimungkinkan untuk mempunyai pelatihan, pengetahuan, kemampuan, karakteristik fisik dan sebagainya. Diagram sebab akibat dapat digunakan untuk menggambarkan semua penyebab utama dan minor.

Selain menggunakan diagram sebab akibat, untuk melakukan pembuatan struktur hirarki juga menggunakan diagram pohon. Diagram pohon dapat digunakan untuk memberi gambaran struktur hirarki yang lebih jelas. Dalam diagram pohon, angka yang dituliskan diatas tiap faktor menandakan tingkat dan nomor faktor pada tingkat tersebut. Sedangkan angka faktor menandakan hubungan interaksi faktor. Seperti terlihat pada gambar 2.4 berikut.



Gambar 2.4 Diagram Pohon

c. Perhitungan Efek Tiap Faktor terhadap Kinerja

Pada tahap ini, digunakan metode *Analytical Hierarchy Process* (AHP). Dengan metode AHP dapat diperoleh bobot setiap faktor terhadap pengukuran kinerja. Semakin besar bobot satu faktor terhadap pengukuran kinerja maka semakin penting pula efek faktor tersebut terhadap kinerja perusahaan. Mengenai AHP lebih lengkap akan dibahas pada subbab 2.3.

2.3. METODE ANALYTICAL HIERARCHY PROCESS

Pada subbab ini dijelaskan tentang metode *Analytical Hierarchy Process* (AHP) sebagai salah satu metode pembobotan faktor pengukuran kinerja, yang terdiri dari deskripsi AHP, manfaat AHP, kelemahan AHP, serta prosedur dalam AHP.

2.3.1. Deskripsi AHP

Analytical Hierarchy Process (AHP) dikembangkan oleh Prof. Thomas L. Saaty, seorang Guru Besar Matematika dari *University of Pittsburgh* pada tahun 1980. Metode ini merupakan alat bantu sistem pendukung keputusan yang dikenal luas untuk menyelesaikan persoalan keputusan multikriteria. Metode ini mensintesis perbandingan pengambil keputusan yang berpasangan pada setiap tingkat hirarki keputusan [SAA80].

Metode ini juga merupakan salah satu bentuk metode pengambilan keputusan dalam menyelesaikan masalah yang kompleks dan tidak terstruktur atau masalah yang tidak memiliki data tertulis yang cukup, seperti permasalahan penentuan alternatif, penyusunan prioritas, pemecahan konflik, pemilihan kebijakan, pengalokasian sumber, penentuan kebutuhan, dan juga pengukuran perfomansi [SAA80]. Dengan AHP ini dapat ditetapkan bobot prioritas relatif setiap elemen keputusan, dimana bobot ini merepresentasikan intensitas preferensi atas suatu elemen keputusan [ARI04].

2.3.2. Manfaat AHP

Manfaat dari metode AHP antara lain adalah [SUM03]:

- Memadukan intuisi, pemikiran, perasaan, dan penginderaan dalam menganalisis pengambilan keputusan.
- Memperhitungkan konsistensi dan penilaian yang telah dilakukan dalam membandingkan faktor-faktor untuk memvalidasi keputusan.
- Kemudahan pengukuran dalam elemen.
- Memungkinkan untuk melakukan perencanaan ke depan (*forward*) dan sebaliknya, menjabarkan masa depan yang ingin dicapai ke masa kini (*backward*).

Salah satu manfaat utama AHP yang membedakan dengan model pengambilan keputusan lainnya adalah tidak ada sarat konsistensi mutlak. Hal ini didasarkan pada kenyataan bahwa keputusan manusia sebagian didasarkan pada logika dan sebagian lagi didasarkan pada unsur bukan logika seperti pengalaman dan intuisi.

2.3.3. Kelemahan AHP

Selain mempunyai kelebihan, AHP juga memiliki kelemahan yaitu [SUM03]:

- Dalam penerapannya harus melibatkan orang-orang yang memiliki pengetahuan yang cukup tentang permasalahan dan tentang metode AHP itu sendiri.
- Untuk melakukan perbaikan keputusan, walaupun kecil, harus dimulai lagi dari tahap awal dengan kebutuhan waktu yang relatif sama.
- Tidak dapat diterapkan pada suatu perbedaan sudut pandang yang sangat tajam atau ekstrim di kalangan responden.

2.3.4. Prosedur AHP

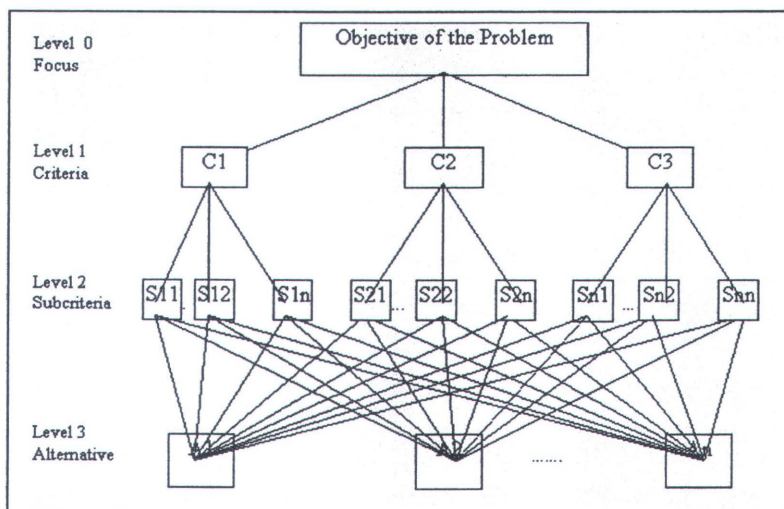
Dalam subbab ini dijelaskan prosedur AHP yang meliputi pembentukan hirarki, perbandingan pasangan, pengecekan konsistensi dan evaluasi [SAA80].

a. Pembentukan Hirarki.

Hirarki dapat membantu untuk menyederhanakan suatu masalah yang rumit menjadi lebih terstruktur. Sebuah hirarki menunjukkan pengaruh tujuan dari tingkat atas sampai pada tingkat yang paling bawah. Hirarki dapat diuraikan menjadi dua jenis, yaitu:

- Hirarki struktural, yaitu suatu pembagian masalah yang rumit ke dalam kelompok-kelompok yang lebih kecil berdasarkan ukuran-ukuran tertentu.
- Hirarki fungsional, yaitu suatu penguraian masalah ke dalam beberapa bagian didasarkan atas hubungan esensialnya.

Pada metode AHP ini, pembentukan hirarki dibuat dengan menggunakan diagram pohon (*tree diagram*) sebagaimana pada gambar 2.5 berikut.



Gambar 2.5 Diagram Hirarki AHP

b. Perbandingan Pasangan.

Prosedur ini melibatkan proses perbandingan berpasangan yang digunakan untuk mempertimbangkan faktor-faktor keputusan atau alternatif-alternatif dengan memperhitungkan hubungan antara faktor dan sub-faktor. Dengan cara dibandingkan antara sejumlah faktor atau alternatif secara berpasangan. Untuk mempermudah perbandingannya, dapat menggunakan tabel matriks berpasangan seperti pada tabel 2.1.

Tabel 2.1 Tabel Matriks Perbandingan Berpasangan

	A1	A2	A3	...	An
A1	1	W1/W2	W1/W3	...	W1/Wn
A2	W2/W1	1	W2/W3	...	W2/Wn
A3	W3/W1	W3/W2	1	...	W3/Wn
...	1	...
An	Wn/W1	Wn/W2	Wn/W3	...	1

Pada tabel 2.1, dimisalkan terdapat n jumlah faktor. Selanjutnya, setiap faktor dibandingkan antara satu faktor dengan faktor lain secara berpasangan. Notasi A1 sampai dengan An menyatakan jenis-jenis indikator yang dibandingkan dalam satu tingkat. Sedangkan notasi W1 sampai dengan Wn menyatakan nilai intensitas kepentingan.

Dalam perbandingan berpasangan ini digunakan angka yang menyatakan intensitas kepentingan. Setiap nilai intensitas kepentingan menunjukkan besarnya pengaruh atau kepentingan suatu faktor terhadap faktor yang lain. Pada tabel 2.2 berikut menjelaskan mengenai intensitas kepentingan antar suatu faktor dibandingkan dengan faktor lain.

Tabel 2.2 Intensitas Kepentingan

Intensitas Kepentingan	Keterangan	Penjelasan
1	Kedua elemen sama penting daripada elemen lainnya	Dua elemen mempunyai pengaruh yang sama besarnya terhadap tujuan
3	Elemen yang satu Sedikit lebih penting daripada elemen lainnya	Pengalaman dan penilaian sedikit menyokong satu elemen dibanding elemen lainnya
5	Elemen yang satu cukup penting daripada elemen lainnya	Pengalaman dan penilaian kuat menyokong satu elemen dibanding elemen lainnya
7	Elemen yang satu sangat penting daripada elemen lainnya	Satu elemen yang kuat disokong dan dominan terlihat dalam praktek
9	Elemen yang satu sangat penting sekali (mutlak) daripada elemen lainnya	Bukti mendukung elemen satu terhadap elemen lain memiliki tingkat penegasan tertinggi yang mungkin terkuat
2, 4, 6, 8	Nilai tengah sebagai refleksi kompromisasi	Nilai ini diberikan bila ada dua kompromi

c. Pengecekan Konsistensi.

Pengecekan konsistensi dilakukan untuk melihat apakah perbandingan berpasangan yang sudah dibuat masih berada di dalam batas kontrol penerimaan atau tidak. Jika ternyata tidak, maka perlu dilakukan kajian ulang untuk menyelidiki apakah konsistensi tersebut dapat diaplikasikan.

Hubungan preferensi yang dikenakan antara dua elemen tidak mempunyai masalah konsistensi relasi. Bila elemen A adalah dua kali lebih penting dari elemen B, maka elemen B adalah setengah lebih penting dari elemen A. Tetapi konsistensi seperti itu tidak selalu berlaku bila terdapat banyak elemen yang harus dibandingkan. Karena keterbatasan kemampuan numerik manusia maka prioritas yang diberikan untuk sekumpulan elemen tidaklah selalu konsisten secara logis.

Dalam prakteknya, konsistensi seperti diatas sulit sekali didapatkan. Suatu matriks perbandingan berpasangan dapat dikatakan konsisten jika dan hanya jika :

$$\lambda_{\max} = n$$

dimana

λ_{\max} : Eigen value terbesar

n : Dimensi dari Matriks A

dan nilainya harus : $\lambda_{\max} \geq n$

Sehingga dihitung suatu parameter *Consistency Index* (CI), dengan rumus sebagai berikut :

$$CI = \frac{\lambda_{\max} - n}{n - 1}$$

Nilai CI tidak akan berarti bila tidak terdapat tolak ukur untuk menyatakan apakah CI menunjukkan suatu matriks yang konsisten. Suatu matriks yang dihasilkan dari perbandingan yang dilakukan secara acak merupakan suatu matriks yang mutlak tidak konsisten. Dari matriks random tersebut didapatkan juga nilai *Consistency Index*, yang disebut dengan *Random Index* (RI). Dengan membandingkan CI dengan RI, maka didapatkan patokan untuk menentukan tingkat konsistensi suatu matriks, yang disebut dengan *Consistency Ratio* (CR), dengan rumus :

$$CR = \frac{CI}{RI}$$

Jika nilai CR dibawah 10 % maka hasil perbandingan dapat diterima. Dan jika melebihi nilai CR tersebut, maka hasil perbandingan tidak dapat diterima. Atau dengan kata lain bahwa telah terjadi ketidak konsistenan suatu nilai

pada saat penentuan prioritas. Jika terjadi ketidak konsistenan maka akan dilakukan pengisian ulang atau dengan cara membuat suatu hirarki baru dibawah kriteria tersebut, agar pengambil keputusan dapat lebih konsisten.

d. Evaluasi.

Evaluasi yang dimaksud adalah evaluasi seluruh proses pembobotan, dimana bobot dari seluruh alternatif harus diketahui. Bobot tersebut harus dilakukan proses normalisasi pada setiap matriks perbandingan pasangan. Alternatif dengan bobot tertinggi adalah alternatif dengan prioritas tinggi, sehingga merupakan alternatif terbaik.

2.4. BALANCED SCORECARD

Pengukuran kinerja merupakan salah satu faktor penting bagi perusahaan. Karena hasil dari pengukuran kinerja, antara lain, dapat dipergunakan sebagai dasar untuk menyusun sistem penghargaan (*reward*) dan penilaian keberhasilan perusahaan [SUM03].

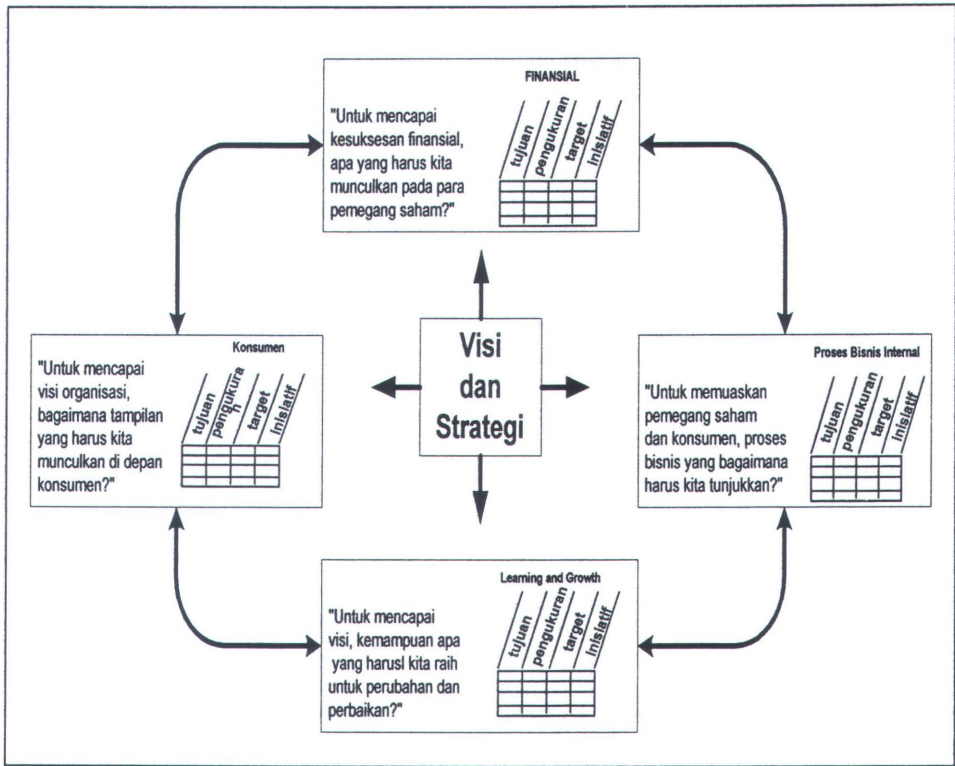
Balanced Scorecard (BSC) merupakan sistem pengukuran kinerja yang mengkombinasikan penilaian keuangan tradisional dengan penilaian non-keuangan yang terkait dengan kepuasan pelanggan (*customer*). Pengertian *balanced* disini diartikan bahwa jika suatu perusahaan atau organisasi menggunakan model tersebut, maka pengukuran kinerja yang dilakukan akan berusaha menyeimbangkan faktor-faktor finansial dan non-finansial, pengukuran jangka panjang (*long term*) dan jangka pendek (*short term*) serta pengukuran kinerja internal dan eksternal [SUM03].

Oleh karena, apa yang diukur oleh suatu organisasi adalah apa yang didapat, maka penting bahwa sistem pengukurannya merefleksikan tujuan dan strategi

organisasi tersebut. Telah diketahui bahwa pengukuran keuangan tradisional seperti ROI, dan EPS tidak lagi mengukur seluruh kemampuan yang dibutuhkan pegawai. Oleh karena itu, BSC berusaha mencapai suatu “keseimbangan” dengan cara mengukur 4 perspektif yang berbeda, yaitu perspektif keuangan, pelanggan, proses bisnis internal, serta pertumbuhan dan pembelajaran [SUM03].

Keuntungan dari metode pengukuran kinerja berdasarkan BSC, adalah karena pendekatan ini berusaha untuk menerjemahkan visi, misi dan tujuan suatu organisasi ke dalam sasaran (*objective*) dan pengukuran kinerja (*performance measure*). Tujuan dan pengukuran kinerja tersebut dibuat strukturnya ke dalam empat perspektif BSC. Kekuatan lain dari model BSC adalah memungkinkan untuk menyusun suatu program peningkatan yang akan membantu organisasi untuk dapat bertahan hidup (*survival*) dalam jangka panjang, yaitu melalui perspektif pertumbuhan dan pembelajaran. Oleh karena tujuan dan pengukuran kinerja ini diturunkan langsung dari visi, misi dan tujuan, maka organisasi yang mempunyai strategi dan misi yang berbeda harus menggunakan pengukuran kinerja yang berbeda pula.

Balanced Scorecard adalah salah satu model sistem pengukuran kinerja untuk mengidentifikasi parameter pengukuran kinerja, yang diturunkan dari visi, misi, dan tujuan ke dalam perspektif-perspektif yang berupa *scorecard-scorecard*. *Scorecard* tersebut antara lain yaitu perspektif finansial, konsumen, proses bisnis internal, dan perspektif tumbuh dan belajar. Di mana ruang lingkup tiap-tiap perspektif seperti yang terlihat pada gambar 2.6, dapat dijelaskan pada subbab berikut [IND02].



Gambar 2.6 Framework dalam Balanced Scorecard untuk menerjemahkan strategi ke dalam proses-proses operasional

Perspektif Finansial

Balanced Scorecard tetap mempertahankan aspek finansial, karena perspektif ini masih tetap diperlukan untuk memberi ringkasan dari konsekuensi ekonomis akibat dari kebijaksanaan-kebijaksanaan yang telah diambil. Tolak ukur dalam bidang finansial mengindikasikan bahwa strategi perusahaan, implementasi dan eksekusi yang dilakukan memberikan kontribusi kepada *bottom-line performance* (kinerja akhir).



Aspek finansial ini erat hubungannya dengan profitabilitas, *Return on Capital*, dan *Economic Added Value*. Tujuan finansial yang lain dapat berupa pertumbuhan penjualan, *cash flow* yang lancar, dsb [IND02].

Perspektif Konsumen

Pada aspek ini, para manajer mengidentifikasi segmen pasar dan konsumen di mana unit kerja akan saling bersaing serta tolak ukur yang akan dipakai untuk mengukur segmen yang diinginkan.

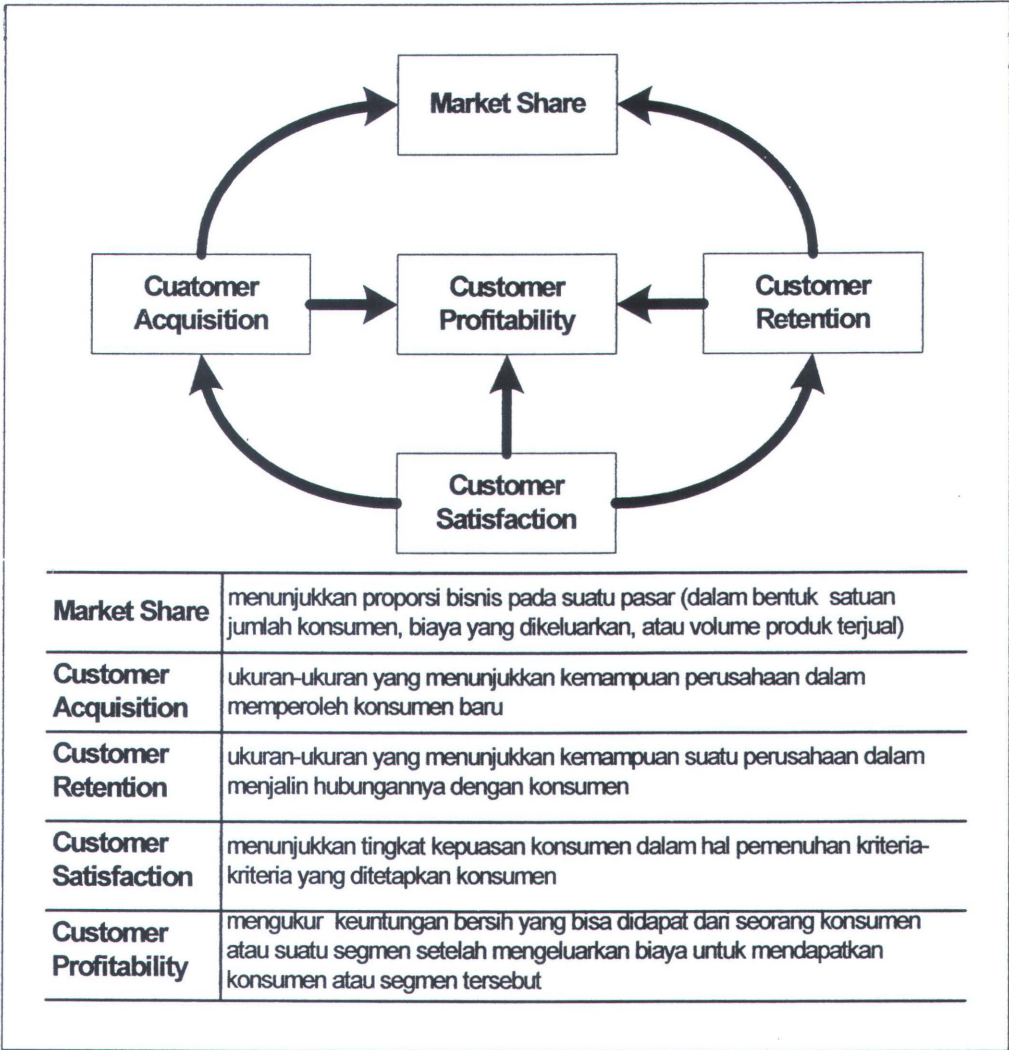
Pengukuran kinerja untuk perspektif konsumen memiliki ukuran-ukuran inti yang direkomendasikan Kaplan & Norton (1996) dan biasanya dapat diimplementasikan pada seluruh tipe organisasi [IND02]. Kelompok ukuran inti tersebut antara lain:

- *market share.*
- *customer retention.*
- *customer acquisition.*
- *customer satisfaction.*
- *customer profitability.*

Ukuran-ukuran inti ini dapat dikelompokkan dalam rantai hubungan sebab-akibat seperti dilihat pada gambar 2.7.

Perspektif Proses Internal

Pada perspektif ini, para eksekutif harus dapat mengidentifikasi *internal business process* yang kritis, yang harus ditingkatkan. Proses-proses ini memungkinkan setiap unit bisnis melakukan hal-hal berikut, antara lain [IND02]:



Gambar 2.7 Perspektif Konsumen-Core Measures

- Mengantarkan proporsi nilai yang akan menarik dan mempertahankan konsumen pada target segmen yang ditetapkan.
- Memberikan kepuasan kepada keinginan para pemegang saham untuk mendapatkan keuntungan finansial yang diharapkan.

Tolak ukur dalam perspektif ini memfokuskan pada proses internal yang akan memberikan dampak yang paling besar pada kepuasan konsumen dan mencapai tujuan finansial dari organisasi.

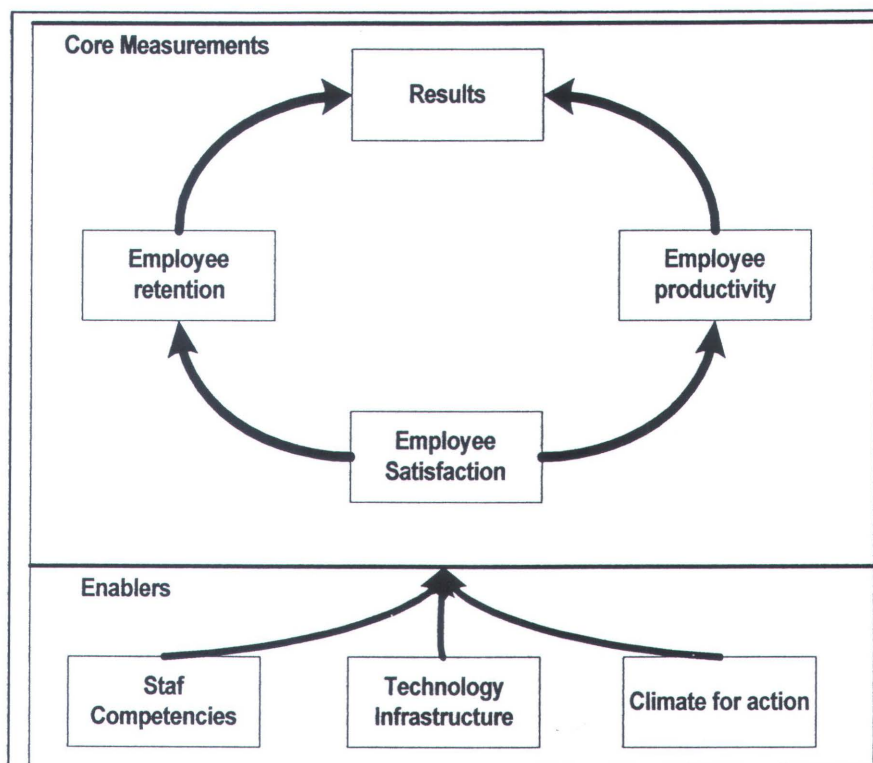
Proses Belajar dan Pertumbuhan

Perspektif keempat dari *scorecard* adalah mengidentifikasi infrastruktur dari organisasi yang harus dibangun untuk menghasilkan pertumbuhan dan perbaikan jangka panjang. Kedua perspektif sebelumnya mengidentifikasi faktor yang paling kritis untuk periode sekarang dan di masa yang akan datang.


Ada tiga prinsip utama dalam perspektif ini, yaitu manusia, sistem dan prosedur organisasi. Ketiga perspektif sebelumnya akan menunjukkan *gap* yang besar dengan kemampuan yang ada pada ketiga aspek tadi dengan kemampuan yang dibutuhkan untuk mencapai kinerja yang diinginkan. Untuk mempersempit *gap* ini, maka perusahaan harus mulai berinvestasi dengan *re-skilling* karyawan, meningkatkan teknologi informasi dan sistem, serta menata ulang prosedur organisasi dan rutinitasnya.

Kaplan dan Norton (1996) telah menemukan tiga ukuran inti yang dapat disesuaikan dengan kondisi perusahaan yang menggunakannya dan bagi perusahaan yang melibatkan *objective employee* dapat menarik beberapa ukuran lainnya dari ketiga ukuran inti tersebut. Ketiga ukuran performansi pekerja ini seperti pada gambar 2.8, adalah sebagai berikut [IND02]:

- *Employee satisfaction.*
- *Employee retention.*
- *Employee productivity.*



Gambar 2.8 Network Pengukuran Perspektif Tumbuh dan Belajar



*Lanjutkan perjuangan...!!
Hidup Mahasiswa...!! Hidup Rakyat...!!*

BAB III
SISTEM PENGUKURAN KINERJA
DI PT MERPATI NUSANTARA AIRLINES

BAB III

SISTEM PENGUKURAN KINERJA

DI PT MERPATI NUSANTARA AIRLINES SURABAYA

Bab ini akan menjelaskan mengenai sistem pengukuran kinerja yang digunakan di PT Merpati Nusantara Airlines Surabaya, yang terdiri dari identifikasi faktor-faktor pengukuran kinerja dan pembobotannya, dan struktur hirarki analitis faktor-faktor pengukuran kinerja.

3.1. IDENTIFIKASI FAKTOR-FAKTOR PENGUKURAN KINERJA

Identifikasi faktor-faktor pengukuran kinerja dan pembobotannya dilakukan pada setiap awal periode pengukuran kinerja. Sebagai sebuah studi kasus, dalam bab ini dibahas tahap-tahap identifikasi dan pembobotan pengukuran kinerja yang digunakan di Divisi MMF (*Merpati Maintenance Facility*) Direktorat Teknik PT Merpati Nusantara Airlines Surabaya.

3.3.1. Identifikasi Visi, Misi, Tujuan dan Strategi Perusahaan

Dalam menjalankan proses bisnisnya MMF selalu memperhatikan visi dan misi yang telah ditetapkan oleh perusahaan, agar segala kegiatan yang dijalankan terarah dan sesuai dengan apa yang diinginkan oleh perusahaan. Di bawah ini dijelaskan visi, misi, tujuan dan strategi dari MMF.

3.3.1.1. Visi, Misi dan Tujuan MMF

Visi MMF yaitu:

“Menjadi pilihan utama perawatan pesawat dan komponen di Indonesia”.

Misi MMF antara lain:

- a. Menjamin kualitas produk.
- b. Menjaga keterandalan pesawat dan komponen.
- c. Menawarkan biaya yang kompetitif.
- d. Mendapatkan keuntungan yang wajar.

Tujuan MMF antara lain:

- a. Menjamin kelayakan udara pesawat, komponen dan kelengkapannya yang dirawat di MMF.
- b. Menyediakan jasa perawatan pesawat, komponen dan kelengkapannya yang berkualitas untuk memuaskan kebutuhan pelanggan.

3.3.1.2. Strategi MMF

Terdapat dua jenis strategi yang digunakan oleh MMF, yaitu:

- a. Pertumbuhan Terkonsentrasi

MMF berkembang dari *core* bisnis maintenance, dengan cara memperluas kapabilitas yang dimiliki.

- b. Diversifikasi Konsentrik

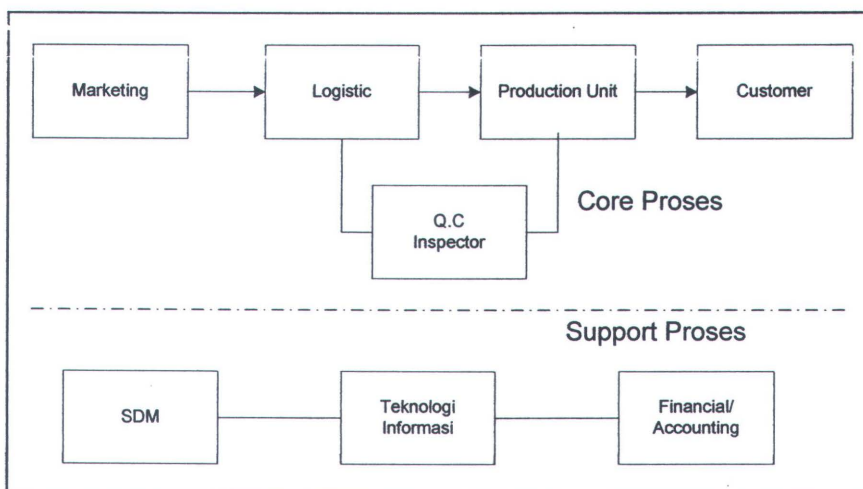
Strategi kedua yang dilakukan oleh MMF adalah dengan cara menambah produk baru, akan tetapi tetap berkaitan dengan yang dimiliki.

3.3.2. Identifikasi Objektif MMF

Pada subbab ini dijelaskan tentang tahap pengidentifikasian pada MMF, meliputi identifikasi proses bisnis, identifikasi objektif awal dan identifikasi objektif final.

3.3.2.1. Identifikasi Proses Bisnis

Aliran proses bisnis utama pada MMF dimulai dari bagian marketing yang berfungsi sebagai pencari dan penerima order, dan berlanjut ke bagian logistik untuk untuk melihat ketersediaan komponen dan fasilitas yang dipergunakan. Setelah dari bagian logistik, dilanjutkan ke bagian *production unit* yang berfungsi sebagai tempat utama proses pemeliharaan pesawat dan komponennya. Setelah urutan kerja (*work order*) selesai dikerjakan, diserahkan kembali ke konsumen. Diantara proses dari bagian logistik ke bagian *production unit*, terdapat *quality control inspector* yang berfungsi sebagai pemelihara kualitas. *quality control inspector* ini menjamin proses dan kualitas komponen sesuai dengan standar. Aliran proses bisnis yang digunakan oleh MMF dapat digambarkan seperti diperlihatkan dalam gambar 3.1 berikut.



Gambar 3.1 Proses Bisnis MMF

3.3.2.2. Identifikasi Objektif Awal MMF

Sebagai permulaan, penentuan objektif awal yang dapat diturunkan dari visi dan misi perusahaan adalah sebagai berikut. Identifikasi objektif awal ini dilakukan dengan metode wawancara dengan pihak manajemen yang terkait dengan masing-masing perspektif, antara lain:

Tabel 3.1 Daftar Narasumber Wawancara Identifikasi Objektif

No	Narasumber	Perspektif
1	Manajer Departemen Keuangan	Perspektif Finansial
2	Manajer Departemen Marketing	Perspektif Konsumen
3	Manajer Departemen Quality Control	Perspektif Proses Internal
4	Manajer Departemen Human Resources	Perspektif Tumbuh dan belajar

Dari data awal hasil wawancara, setelah diseleksi didapatkan objektif-objektif awal divisi MMF. Secara umum sebagaimana konsep *Balanced Score Card*, objektif-objektif awal ini dikelompokkan menjadi empat jenis perspektif. Selengkapnya seperti terlihat pada tabel 3.2 berikut.

Tabel 3.2 Hasil Penentuan Objektif Awal

Perspektif	Objectives
Perspektif Finansial	a. Meningkatkan profit b. Meningkatkan akuntabilitas
Perspektif Konsumen	a. Meningkatkan Kepuasan pelanggan b. Meningkatkan <i>market share</i>
Perspektif Proses Internal	a. Meningkatkan mutu b. Meminimalkan problem operasi c. Meningkatkan produktivitas shop
Perspektif Tumbuh dan belajar	a. Meningkatkan kinerja karyawan b. Meningkatkan Infrastruktur teknologi

3.3.2.3. Identifikasi Objektif Final

Dari pemunculan objektif awal yang didapat dari data pertama, perlu dilakukan pengujian apakah objektif yang dimunculkan sesuai dengan kebutuhan. Agar lebih terfokus, maka untuk tahap identifikasi objektif final dilakukan dengan metode kuisisioner terhadap nara sumber pihak manajemen terkait. Dengan metode kuisisioner, maka hasil yang didapat menjadi lebih terarah daripada metode wawancara. Hasil identifikasi objektif final dirangkum pada tabel 3.3.

Tabel 3.3 Penyesuaian Objektif Final

Perspektif	Objektif	A	B	C	D	Jumlah
Perspektif finansial	a. Meningkatkan profit	4	3	3	3	13
	b. Meningkatkan akuntabilitas keuangan	4	3	3	3	13
Perspektif konsumen	a. Meningkatkan kepuasan pelanggan	4	4	4	4	16
	b. Meningkatkan market share	3	3	3	3	12
Perspektif proses internal	a. Meningkatkan mutu	4	4	4	4	16
	b. Meminimalkan problem operasi	3	3	4	3	13
	c. Meningkatkan produktivitas shop	3	4	4	3	14
Perspektif tumbuh dan belajar	a. Meningkatkan kinerja karyawan	3	3	3	4	13
	b. Meningkatkan infrastruktur teknologi	3	3	3	3	12

Makna dari tabel diatas dapat dijelaskan sebagai berikut. Tabel tersebut diperoleh dari hasil kuisisioner yang diberikan kepada empat orang manajer yaitu manajer A, B, C, dan manajer D, sehingga terdapat empat data penilaian untuk masing-masing objektif final. Pengisi kuisisioner memberikan nilai dengan angka 1 (satu) sampai 4 (empat) pada masing-masing objektif final. Nilai 1 untuk sangat tidak setuju, nilai 2 untuk tidak setuju, nilai 3 untuk setuju, dan nilai 4 untuk sangat setuju.

Berdasarkan nilai penyesuaian diatas, ternyata rata-rata nilai setiap objektif lebih besar sama dengan nilai 12, yang berarti jika dirata-rata, setiap responden minimal menyetujui objektif yang diajukan. Sehingga keseluruhan objektif akan

dimasukkan pada pengolahan selanjutnya. Jika misalnya ternyata ada salah satu objektif yang bernilai kurang dari nilai 12, maka objektif tersebut tidak dimasukkan kedalam pengolahan identifikasi KPI [ARI04].

3.3.3. Identifikasi Key Performance Indicator (KPI)

Dari objektif-objektif diatas, KPI dapat dicari melalui diskusi, wawancara dan dokumen internal yang menjelaskan sistem dalam perusahaan. Pemahaman tentang keseluruhan sistem perusahaan berguna sebagai pegangan peneliti untuk mengarahkan identifikasi indikator-indikator yang tepat. Pemahaman tentang sistem perusahaan dapat dituangkan pada diagram pemetaan (*mapping*) proses bisnis yang menggambarkan keterkaitan dan keseluruhan aktivitas perusahaan.

Pemetaan KPI dengan menjelaskan proses bisnis perusahaan secara menyeluruh seperti disebutkan di atas, akan mempermudah perancang untuk mengidentifikasi indikator-indikator kunci yang mencerminkan pencapaian objektif pada setiap proses bisnis di dalam perusahaan. Karena pemetaan ini ditekankan pada perspektif proses bisnis internal, maka indikator-indikator pada perspektif lain yang tidak tercantum bukan berarti tidak penting. Indikator-indikator lain yang dianggap penting, tetapi tidak berhubungan langsung dengan proses bisnis internal diikutsertakan berdasarkan kebutuhan dan objektif yang ada.

Berdasarkan objektif yang telah didapatkan, KPI dapat dicari melalui diskusi dengan nara sumber dan dokumen internal yang menjelaskan sistem di divisi MMF PT Merpati Nusantara Airlines Surabaya. Keseluruhan KPI yang berlaku pada periode tahun 2004 dapat dilihat pada tabel 3.4 berikut [ARI04].

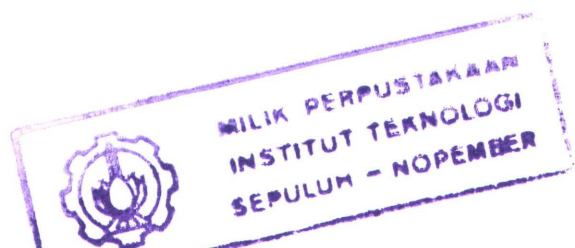
Tabel 3.4 Hasil Identifikasi KPI

Perspektif	Objektif	Key Performance Indicator
Finansial	F.1. Meningkatkan Profit	F.1.1. Jumlah profit
		F.1.2. <i>Cost Reduction</i>
	F.2. Meningkatkan akuntabilitas	F.2.1. Frekuensi Pelaporan
		F.2.2. <i>Current Ratio</i>
		F.2.3. <i>Quick Ratio</i>
Konsumen	C.1. Meningkatkan kepuasan pelanggan	C.1.1. Kepuasan pelanggan
		C.1.2. Prosentase order terkomplain
		C.1.3. Nominal order terkomplain
	C.2. Meningkatkan <i>Market Share</i>	C.2.1. Prosentase <i>market share</i> yang diraih
		C.2.2. Pertumbuhan pelanggan
Proses Internal	I.1. Meningkatkan mutu	I.1.1. <i>Post Inspection Defect</i>
		I.1.2. Kualitas supplier
		I.1.3. Jumlah kecelakaan kerja
	I.2. Meminimalkan problem operasi	I.2.1. <i>Turn around Time</i>
		I.2.2. Kesalahan pembelian
		I.2.3. Kesalahan pengiriman
	I.3. Meningkatkan produktivitas shop	I.3.1. Component Shop
		I.2.4. Engine & Propeller Shop
		I.2.5. Heavy Maintenance
Proses Tumbuh dan Belajar	L.1. Meningkatkan kinerja pegawai	L.1.1. <i>Employee Training Hours</i>
		L.1.2. <i>Labour Target Achievement</i>
		L.1.3. Kesejahteraan karyawan
		L.1.4. Jumlah karyawan dg keluhan kerja
	L.2. Meningkatkan infrastruktur teknologi	L.2.1. Rasio Jumlah komputer dg pegawai
		L.2.2. Jumlah klaim sistem informasi

3.2. STRUKTUR HIRARKI ANALITIS FAKTOR-FAKTOR

PENGUKURAN KINERJA DAN PELEVELANNYA

Berdasarkan model QMPMS, setelah diperoleh KPI-KPI yang digunakan pada divisi MMF periode tahun 2004, dilakukan pembuatan struktur hirarki pada KPI-KPI tersebut. Proses pembuatan struktur hirarki KPI dilakukan dengan menggunakan diagram pohon. Tujuannnya adalah agar



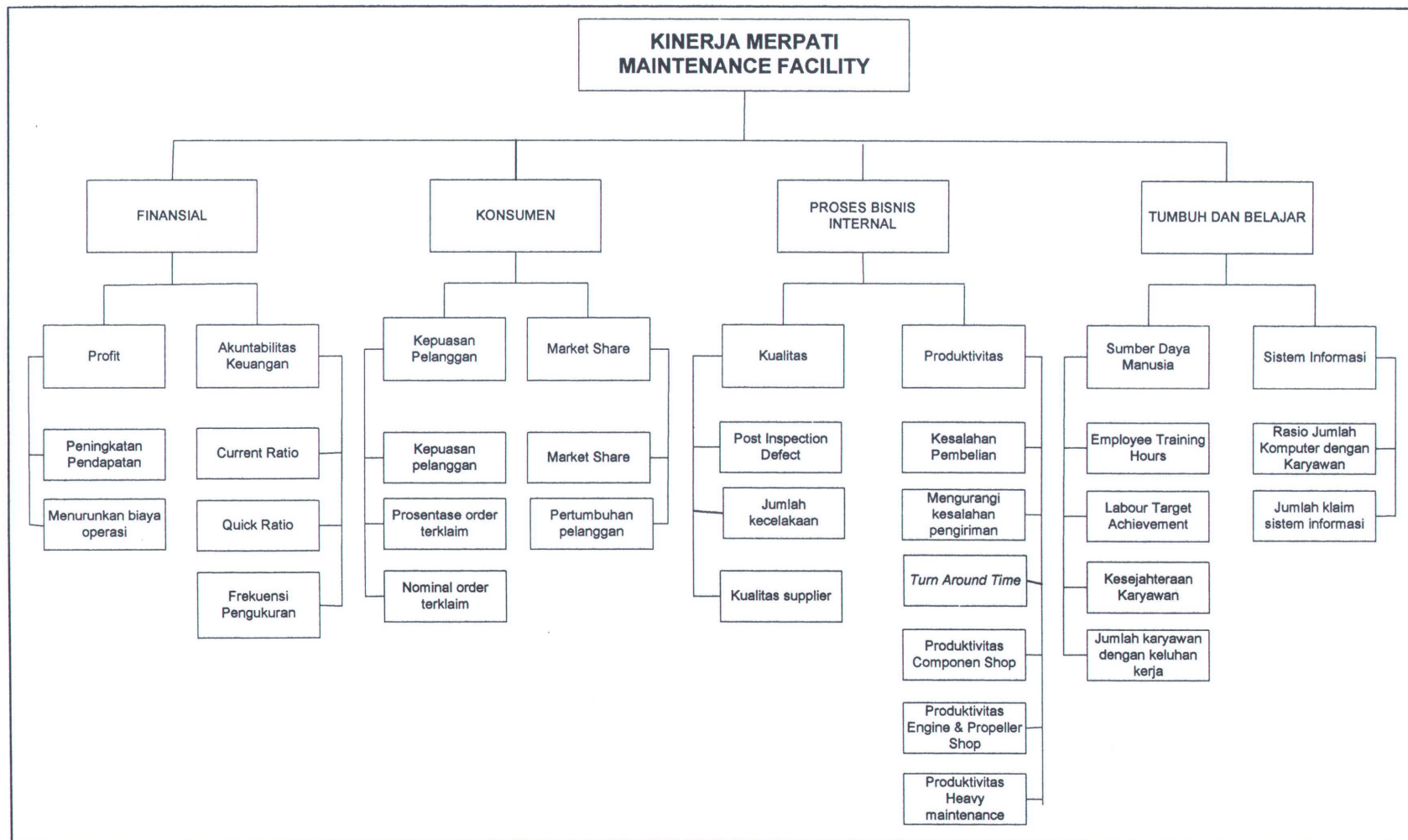
dapat terlihat secara hirarki struktur dari KPI-KPI yang telah diidentifikasi, lengkap dengan tingkat (*level*) setiap KPI.

Pembuatan diagram pohon dibuat berdasarkan objektif-objektif yang telah diidentifikasi pada tahap sebelumnya. Pembuatan diagram pohon diawali dari tingkat yang paling atas, yang pada *Balanced Score Card* untuk tingkat paling atas ini disebut dengan perspektif. Pada penempatan perspektif ini terdapat empat macam, yaitu perspektif finansial, perspektif konsumen, perspektif proses bisnis internal, serta perspektif tumbuh dan belajar.

Pada tingkat kedua, terdapat KPI-KPI turunan yang merupakan penjabaran dari KPI-KPI pada tingkat pertama. KPI-KPI pada tingkat kedua ini disebut dengan kriteria. Sesuai dengan hasil identifikasi, pada divisi MMF ini terdapat delapan buah kriteria.

Pada tingkat ketiga, terdapat KPI-KPI turunan yang merupakan penjabaran dari KPI-KPI pada tingkat kedua. KPI-KPI pada tingkat ketiga ini disebut dengan indikator. Sesuai dengan hasil identifikasi, pada divisi MMF ini terdapat 25 buah indikator.

Selengkapnya bagan struktur hirarki dapat dilihat pada gambar 3.2 berikut.



Gambar 3.2 Struktur Hirarki KPI

3.3. PEMBOBOTAN

Setelah seluruh KPI perusahaan dapat diidentifikasi dan distrukturkan secara hirarkis, tahap selanjutnya adalah melakukan pembobotan KPI untuk mengetahui kontribusi atau pengaruh masing-masing indikator terhadap kinerja perusahaan secara keseluruhan. Metode pembobotan yang digunakan adalah metode *Analytical Hierarchy Process*.

3.3.1. Pembobotan Perspektif

Pembobotan perspektif digunakan dengan menggunakan matriks perbandingan berpasangan. Untuk pembobotan pada level teratas atau yang disebut dengan perspektif menghasilkan nilai bobot seperti pada tabel 3.5 berikut.

Tabel 3.5 Hasil Perhitungan Nilai Bobot Perspektif

Perspektif	Nilai Bobot
Finansial	0,233
Konsumen	0,217
Proses bisnis internal	0,487
Tumbuh dan belajar	0,162

Hasil yang didapat dari pembobotan didapatkan yang mempunyai nilai bobot yang paling tinggi yaitu proses bisnis internal dengan bobot nilai 0,487, selanjutnya perspektif finansial dengan nilai bobot 0,233, disusul dengan perspektif konsumen dengan nilai bobot 0,217, dan terakhir adalah perspektif tumbuh dan belajar dengan nilai bobot 0,162.

3.3.2. Pembobotan pada Kriteria

Pembobotan pada masing-masing kriteria juga menggunakan matriks perbandingan berpasangan. Untuk pembobotan pada level kedua atau yang disebut dengan kriteria menghasilkan nilai bobot sebagai berikut:

a. Perspektif Finansial

Tabel 3.6 Hasil Perhitungan Nilai Bobot Perspektif Finansial

Kriteria	Nilai Bobot
Profitability	0,875
Akuntabilitas keuangan	0,125

b. Perspektif Konsumen

Tabel 3.7 Hasil Perhitungan Nilai Bobot Perspektif Konsumen

Kriteria	Nilai Bobot
Kepuasan pelanggan	0,833
Market share	0,167

c. Perspektif Proses Bisnis Internal

Tabel 3.8 Hasil Perhitungan Nilai Bobot Perspektif Proses Bisnis Internal

Kriteria	Nilai Bobot
Kualitas	0,875
Produktivitas	0,125

d. Perspektif Tumbuh dan belajar

Tabel 3.9 Hasil Perhitungan Nilai Bobot Perspektif Tumbuh dan Belajar

Kriteria	Nilai Bobot
SDM	0,75
Sistem Informasi	0,25

3.3.3. Pembobotan pada Indikator

Setelah pembobotan pada perspektif dan kriteria, tahap selanjutnya adalah melakukan pembobotan untuk mencari bobot dari setiap indikator.

a. Perspektif Finansial

Perspektif finansial mempunyai dua kriteria, yaitu profit dan akuntabilitas.

Nilai bobot indikator dari masing-masing kriteria adalah sebagai berikut :

- Kriteria profit

Tabel 3.10 Hasil Perhitungan Nilai Bobot Kriteria Profit

Indikator	Nilai Bobot
F.1.1. Jumlah <i>profit</i>	0,25
F.1.2. <i>Cost reduction</i>	0,75

- Kriteria akuntabilitas

Tabel 3.11 Hasil Perhitungan Nilai Bobot Kriteria Akuntabilitas

Indikator	Nilai Bobot
F.2.1. Frekuensi pelaporan	0,33
F.2.2. <i>Current ratio</i>	0,33
F.2.3. <i>Quick ratio</i>	0,33

b. Perspektif Konsumen

Perspektif konsumen mempunyai dua kriteria, yaitu kepuasan pelanggan dan *market share*. Nilai bobot dari masing-masing indikator pada setiap kriteria adalah sebagai berikut:

- Kriteria kepuasan pelanggan

Tabel 3.12 Hasil Perhitungan Nilai Bobot Kriteria Kepuasan Pelanggan

Indikator	Nilai Bobot
C.1.1. Kepuasan pelanggan	0,33
C.1.2. Prosentase <i>order</i> terkomplain	0,33
C.1.3. Nominal <i>order</i> terkomplain	0,33

- Kriteria meningkatkan *market share*

Tabel 3.13 Hasil Perhitungan Nilai Bobot Meningkatkan Market Share

Indikator	Nilai Bobot
C.2.1. Prosentase <i>market share</i> yang diraih	0,5
C.1.2. Pertumbuhan pelanggan	0,5

c. **Perspektif Proses Bisnis Internal**

Prespektif bisnis internal mempunyai dua kriteria, yaitu kualitas dan produktivitas. Nilai bobot dari masing-masing kriteria adalah sebagai berikut :

- Kriteria kualitas

Tabel 3.14 Hasil Perhitungan Nilai Bobot Kriteria Kualitas

Indikator	Nilai Bobot
I.1.1 <i>Post Inspection defect</i>	0,487
I.1.2. Jumlah kecelakaan	0,078
I.1.3. Kualitas <i>supplier</i>	0,043

- Kriteria Produktivitas

Tabel 3.15 Hasil Perhitungan Nilai Bobot Kriteria Produktivitas

Indikator	Nilai Bobot
I.2.1. Kesalahan pembelian	0,283
I.2.2. Kesalahan pengiriman	0,065
I.2.3. <i>Turn Around Time</i>	0,414
I.2.4. <i>Component shop</i>	0,079
I.2.5. <i>Engine & propeller Shop</i>	0,079
I.2.6. <i>Heavy Maintenance</i>	0,079

d. **Prespektif Tumbuh dan Belajar**

Prespektif tumbuh dan belajar mempunyai dua kriteria, yaitu Sumber Daya Manusia dan Sistem Informasi. Nilai bobot dari masing-masing kriteria adalah sebagai berikut :

- Kriteria SDM

Tabel 3.16 Hasil Perhitungan Nilai Bobot Kriteria SDM

Indikator	Nilai Bobot
L.1.1. <i>Employee Training Hours</i>	0,120
L.1.2. Labour Target Achievement	0,360
L.1.3. Kesejahteraan karyawan	0,213
L.1.4. Jumlah karyawan dengan keluhan kerja	0.106

- Kriteria Sistem Informasi

Tabel 3.17 Hasil Perhitungan Nilai Bobot Kriteria Sistem Informasi


Indikator	Nilai Bobot
L.2.1 Rasio Jumlah komputer dengan pegawai	0,842
L.2.2. Jumlah klaim sistem informasi	0,176

Dari semua pembobotan KPI, mulai dari perspektif, selanjutnya pembobotan kriteria dan pembobotan indikator, dapat dirangkum seperti pada tabel 3.18 berikut.

Tabel 3.18. Hasil Perhitungan Nilai Bobot Seluruh KPI

Perspektif	Kriteria	Indikator	Nilai Bobot
Finansial	F.1 Meningkatkan profit	F.1.1. Jumlah <i>profit</i>	0,25
		F.1.2. <i>Cost reduction</i>	0,75
	F.2.Meningkatkan akuntabilitas	F.2.1. Frekuensi pelaporan	0,33
		F.2.2. <i>Current ratio</i>	0,33
		F.2.3. <i>Quick ratio</i>	0,33
Konsumen	C.1. Meningkatkan kepuasan pelanggan	C.1.1. Kepuasan pelanggan	0,33
		C.1.2. Prosentase <i>order</i> terkomplain	0,33
		C.1.3. Nominal <i>order</i> terkomplain	0,33
	C.2. Meningkatkan <i>market share</i>	C.2.1. Prosentase <i>market share</i> yang diraih	0,5
		C.2.2. Pertumbuhan pelanggan	0,5

Perspektif	Kriteria	Indikator	Nilai Bobot
Proses Internal	I.1. Meningkatkan kualitas	I.1.1 <i>Post Inspection defect</i>	0,487
		I.1.2. Jumlah kecelakaan	0,078
		I.1.3. Kualitas <i>supplier</i>	0,043
	I.2. Meningkatkan produktivitas	I.2.1. Kesalahan pembelian	0,283
		I.2.2. Kesalahan pengiriman	0,065
		I.2.3. <i>Turn Around Time</i>	0,414
		I.2.4. <i>Component shop</i>	0,079
		I.2.5. <i>Engine & propeller Shop</i>	0,079
		I.2.6. <i>Heavy Maintenance</i>	0,079
Proses Tumbuh dan Belajar	L.1. Meningkatkan kinerja pegawai	L.1.1. <i>Employee Training Hours</i>	0,120
		L.1.2. <i>Labour Target Achievement</i>	0,360
		L.1.3. Kesejahteraan karyawan	0,213
		L.1.4. Jumlah karyawan dg keluhan kerja	0,106
	L.2. Meningkatkan infrastruktur teknologi	L.2.1. Rasio Jumlah komputer dg pegawai	0,842
		L.2.2. Jumlah klaim sistem informasi	0,176



*No ones perfect,
But we can learn to become perfect...!!*

BAB IV

DESAIN PERANGKAT LUNAK

BAB IV

DESAIN PERANGKAT LUNAK

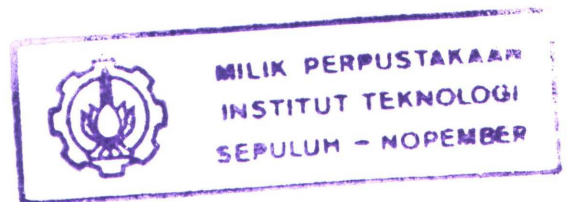
Bab ini menjelaskan tentang tahapan proses desain perangkat lunak. Desain dilakukan dengan menggunakan konsep *Visual Modeling*, yaitu konsep atau cara berpikir tentang suatu permasalahan dengan menggunakan pemodelan visual yang berguna untuk memahami kebutuhan, komunikasi, dokumentasi, dan desain dari perangkat lunak.

Unified Modeling Language (UML) yang menerapkan konsep *visual modeling* menyediakan notasi-notasi yang sangat ampuh untuk digunakan mulai dari fase analisis, desain dan implementasi. Notasi-notasi ini berperan penting dalam menerjemahkan dan memahami berbagai proses yang ada pada suatu organisasi [QUA98].

Pada bab ini, fase desain dibagi menjadi tiga tahap, yaitu desain data, desain proses dan desain antarmuka.

4.1. DESAIN DATA

Tujuan dari desain data adalah untuk mempersiapkan data yang akan digunakan dalam implementasi perangkat lunak. Pada desain data ini, tidak dibahas menggunakan desain data masukan, desain data proses, dan desain data luaran. Hal ini karena pada perangkat lunak ini, data masukan suatu *usecase* nantinya akan menjadi data proses *usecase* lain, ataupun data luaran dari suatu *usecase* akan menjadi masukan bagi *usecase* lain.



Pada data model diagram yang dibuat memiliki sepuluh buah tabel. Penjelasan lebih detail dari tabel-tabel yang digunakan adalah sebagai berikut:

a. Desain Tabel Pengguna

Tabel ini untuk menyimpan data pengguna yang mempunyai hak menggunakan sistem ini, berikut juga hak aksesnya (*role*), dan departemen dari pengguna yang bersangkutan. *Field-field* pada tabel TUser dapat dilihat pada tabel 4.1.

Tabel 4.1 Penjelasan Tabel TUser

Nama Field	Penjelasan
Usr_ID	Berisi data ID karyawan yang punya hak akses. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel TUser. Otomatis digenerate dari JavaBeans TUser. Char(7).
Dept_ID	Berisi data ID Departemen dimana pengguna yang bersangkutan bertugas. <i>Field</i> ini merupakan <i>foreign key</i> dari tabel TDept. Otomatis digenerate dari JavaBeans. Char(7).
Usr_Username	Berisi data nama pengguna karyawan. Varchar(11).
Usr_Password	Berisi data kata kunci dari pengguna. Varchar(11).
Usr_NIK	Berisi data NIK (Nomor Induk Karyawan) pengguna karyawan. Varchar(9).
Usr_Mail	Berisi data <i>e-mail</i> dari pengguna. Varchar(35).
Usr_Role	Berisi data hak akses pengguna terhadap perangkat lunak KPI. Numeric(9).

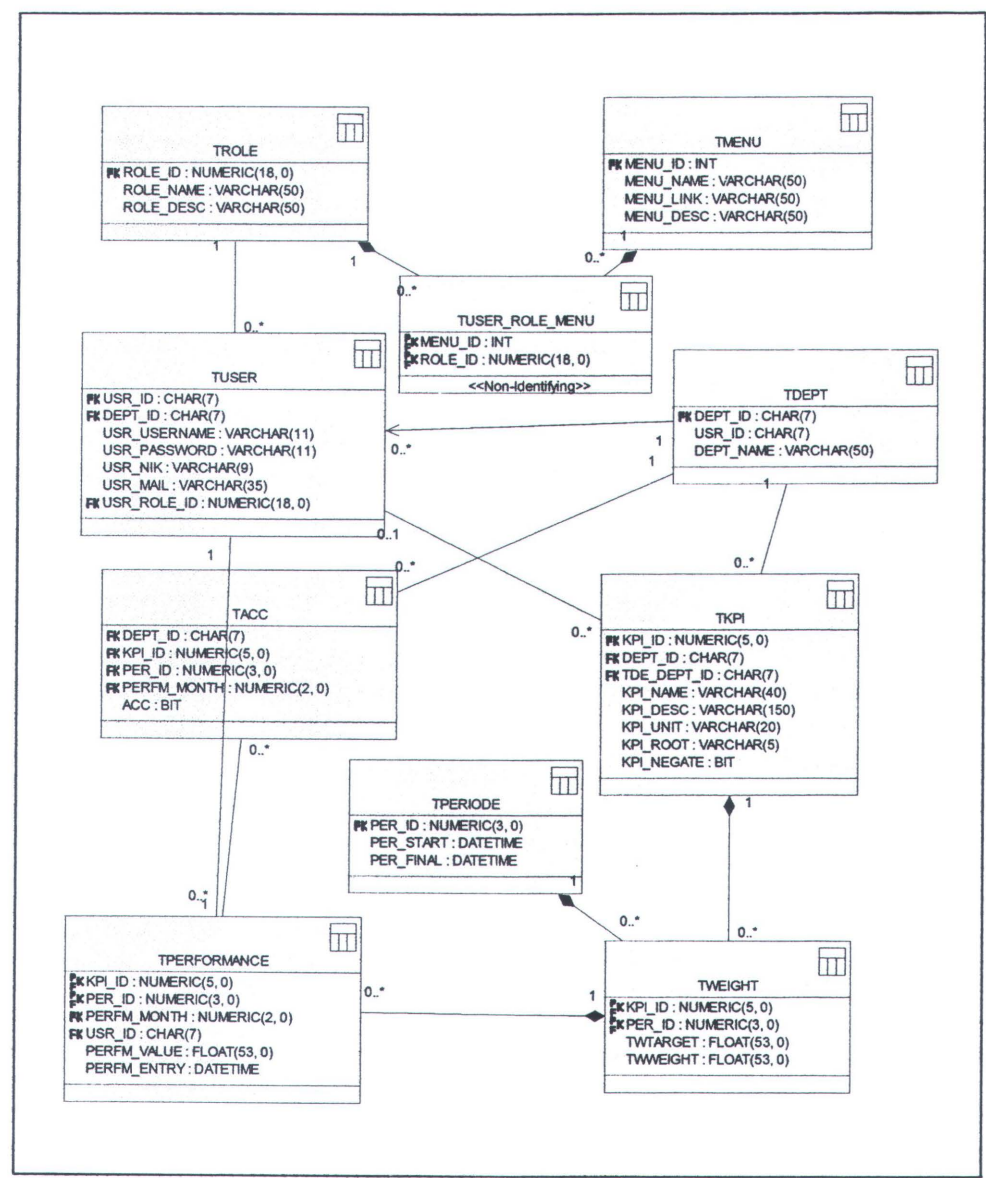
b. Desain Tabel Departemen

Tabel ini untuk menyimpan data departemen yang ada di PT Merpati Nusantara Airlines Surabaya. *Field-field* pada tabel TDept dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Penjelasan Tabel TDept

Nama Field	Penjelasan
Dept_ID	Berisi data ID dari departemen yang bersangkutan. Field ini berfungsi sebagai <i>primary key</i> dari tabel ini. Char(7).
Usr_ID	Berisi data ID karyawan penanggung jawab atau

Dalam bagian ini akan dijelaskan mengenai desain basis data perangkat lunak yang akan dibuat dengan menggunakan *Rational Rose 2000*. Pada gambar 4.1 berikut ini menjelaskan tentang desain *data model diagram* perangkat lunak.



Gambar 4.1 Data Model Diagram

Pada data model diagram yang dibuat memiliki sepuluh buah tabel. Penjelasan lebih detail dari tabel-tabel yang digunakan adalah sebagai berikut:

a. Desain Tabel Pengguna

Tabel ini untuk menyimpan data pengguna yang mempunyai hak menggunakan sistem ini, berikut juga hak aksesnya (*role*), dan departemen dari pengguna yang bersangkutan. *Field-field* pada tabel TUser dapat dilihat pada tabel 4.1.

Tabel 4.1 Penjelasan Tabel TUser

Nama Field	Penjelasan
Usr_ID	Berisi data ID karyawan yang punya hak akses. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel TUser. Otomatis digenerate dari JavaBeans TUser. Char(7).
Dept_ID	Berisi data ID Departemen dimana pengguna yang bersangkutan bertugas. <i>Field</i> ini merupakan <i>foreign key</i> dari tabel TDept. Otomatis digenerate dari JavaBeans. Char(7).
Usr_Username	Berisi data nama pengguna karyawan. Varchar(11).
Usr_Password	Berisi data kata kunci dari pengguna. Varchar(11).
Usr_NIK	Berisi data NIK (Nomor Induk Karyawan) pengguna karyawan. Varchar(9).
Usr_Mail	Berisi data <i>e-mail</i> dari pengguna. Varchar(35).
Usr_Role	Berisi data hak akses pengguna terhadap perangkat lunak KPI. Numeric(9).

b. Desain Tabel Departemen

Tabel ini untuk menyimpan data departemen yang ada di PT Merpati Nusantara Airlines Surabaya. *Field-field* pada tabel TDept dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Penjelasan Tabel TDept

Nama Field	Penjelasan
Dept_ID	Berisi data ID dari departemen yang bersangkutan. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel ini. Char(7).
Usr_ID	Berisi data ID karyawan penanggung jawab atau

Nama Field	Penjelasan
	<i>department head</i> dari departemen yang bersangkutan. Field merupakan <i>foreign key</i> dari tabel TUser. Char(7).
Dept_Name	Berisi data nama dari departemen yang bersangkutan. Varchar(50).

c. Desain Tabel Periode

Tabel ini untuk menyimpan data jangka waktu atau periodisasi dari pengukuran kinerja yang sedang digunakan. *Field-field* pada tabel TPeriode dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Penjelasan Tabel TPeriode

Nama Field	Penjelasan
Per_ID	Berisi data ID periode setiap pengukuran kinerja. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel Periode. Numeric(5).
Per_Start	Berisi data tanggal diawalinya periode pengukuran kinerja. Datetime(8).
Per_Final	Berisi data tanggal berakhirnya periode pengukuran kinerja. Datetime(8).

d. Desain Tabel KPI

Tabel ini untuk menyimpan data faktor-faktor pengukuran kinerja (*key performance indicators*) yang berlaku dalam sistem. *Field-field* pada tabel TKPI dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Penjelasan Tabel TKPI

Nama Field	Penjelasan
KPI_ID	Berisi data ID indikator. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel KPI. Numeric(5).
Dept_ID	Berisi ID dari departemen yang mengisi indikator yang bersangkutan. <i>Field</i> ini merupakan <i>foreign key</i> dari tabel TDept. Char(7).

Nama Field	Penjelasan
Tde_Dept_ID	Berisi ID Pengguna yang meng- <i>accountability</i> (Acc.) indikator yang bersangkutan. <i>Field</i> ini merupakan <i>foreign key</i> dari tabel TUser. Char(7).
KPI_Name	Berisi data nama indikator. Varchar(50).
KPI_Desc	Berisi data penjelasan indikator. Allows Null. Varchar(150).
KPI_Unit	Berisi data satuan dari indikator yang bersangkutan. Varchar(20).
KPI_Root	Berisi data <i>parent</i> dari indikator yang bersangkutan. Varchar(5).
KPI_Negate	Berisi data sifat indikator, apakah sifatnya berkebalikan atau tidak. Bit(1).

e. Desain Tabel Bobot

Tabel ini untuk menyimpan data bobot dan target indikator yang digunakan dalam KPI. *Field-field* pada tabel TWeight dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Penjelasan Tabel TWeight

Nama Field	Penjelasan
Per_ID	Berisi data ID periode. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel TWeight dan juga <i>foreign key</i> dari tabel TPeriode. Numeric(5).
KPI_ID	Berisi data ID indikator. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel Tweight dan juga <i>foreign key</i> dari tabel TKPI. Numeric(5).
TWTarget	Berisi data target untuk indikator dan periode yang bersangkutan. Float(8).
TWWeight	Berisi data bobot untuk indikator dan periode yang bersangkutan. Float(8).

f. Desain Tabel Hasil Pencapaian

Tabel ini untuk menyimpan data hasil pengukuran kinerja perusahaan, yaitu data bulan pelaporan kinerja, nilai pencapaian kinerja dan waktu pemasukan pelaporan kinerja. *Field-field* pada tabel TPerformance dapat dilihat pada tabel 4.6 berikut.

Tabel 4.6 Penjelasan Tabel TPerformance

Nama Field	Penjelasan
KPI_ID	Berisi data ID indikator. <i>Field</i> ini berfungsi sebagai <i>foreign key</i> dari tabel KPI. Numeric(5).
Usr_ID	Berisi data pengguna <i>ID</i> . <i>Field</i> ini berfungsi sebagai <i>foreign key</i> dari tabel TUser. Untuk mencatat ID pengguna yang mengisi pencapaian kinerja. Char(7).
Per_ID	Berisi data ID periode setiap pengukuran kinerja. <i>Field</i> ini berfungsi sebagai <i>foreign key</i> dari tabel TPeriode. Numeric(5).
Perfm_Month	Berisi data <i>datetime</i> ketika data diinputkan. Merupakan <i>Primary Key</i> dari tabel Tperformance. Numeric(5).
Perfm_Value	Berisi data hasil pencapaian suatu KPI. Float(8).
Perfm_Entry	Berisi data tentang tanggal pemasukan pencapaian suatu KPI. Datetime(8).

g. Desain Tabel Accountability

Tabel ini untuk menyimpan data proses *Accountability*. *Field-field* pada tabel TAcc dapat dilihat pada tabel 4.7 berikut.

Tabel 4.7 Penjelasan Tabel TAcc

Nama Field	Penjelasan
KPI_ID	Berisi data ID indikator. <i>Field</i> ini berfungsi sebagai <i>foreign key</i> dari tabel KPI. Numeric(5).
Per_ID	Berisi data ID periode setiap pengukuran kinerja. <i>Field</i> ini berfungsi sebagai <i>foreign key</i> dari tabel TPeriode. Numeric(5).
Perfm_Month	Berisi data bulan data Pencapaian yang telah di-Acc. Merupakan <i>Foreign Key</i> dari tabel Tperformance. Numeric(5).
Dept_ID	Berisi data hasil pencapaian suatu KPI. Char(7).
Acc	Berisi data dalam bit apakah laporan pencapaian kinerja sudah di Acc atau belum. Bit(1).

h. Desain Tabel Hak Akses

Tabel ini untuk menyimpan data jenis-jenis *role* pada perangkat lunak. *Field-field* pada tabel TRole dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 Penjelasan Tabel TRole

Nama Field	Penjelasan
Role_ID	Berisi data ID <i>role</i> . <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel TRole. Numeric(9).
Role_Name	Berisi data nama <i>role</i> . Varchar(50).
Role_Desc	Berisi data deskripsi setiap jenis <i>role</i> . Varchar(50).

i. Desain Tabel Menu

Tabel ini untuk menyimpan data jenis-jenis menu yang ada pada aplikasi beserta keterangan dan alamat *link*-nya. Tabel ini bermanfaat bagi pengembangan fasilitas menu perangkat lunak. *Field-field* pada tabel TMenu dapat dilihat pada tabel 4.9 berikut.

Tabel 4.9 Penjelasan Tabel TMenu

Nama Field	Penjelasan
Menu_ID	Berisi data ID Menu. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel Tmenu. Int(4).
Menu_Name	Berisi data nama menu yang tersedia pada aplikasi. Varchar(50).
Menu_Link	Berisi data alamat link kepada file utama setiap menu. Varchar(50).
Menu_Desc	Berisi data keterangan menu. Varchar(50).

j. Desain Tabel Menu Hak Akses

Tabel ini merupakan tabel hasil relasi *many to many* antara tabel TRole dan TMenu. Tabel ini untuk menyimpan data *foreign key* tabel Trole dan Tmenu. *Field-field* pada tabel TUserRoleMenu dapat dilihat pada tabel 4.10 berikut.

Tabel 4.10 Penjelasan Tabel TMenu

Nama Field	Penjelasan
Menu_ID	Berisi data ID Menu. <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel Tmenu. Int(4).
Role_ID	Berisi data ID <i>role</i> . <i>Field</i> ini berfungsi sebagai <i>primary key</i> dari tabel TRole. Numeric(9).



4.2. DESAIN PROSES

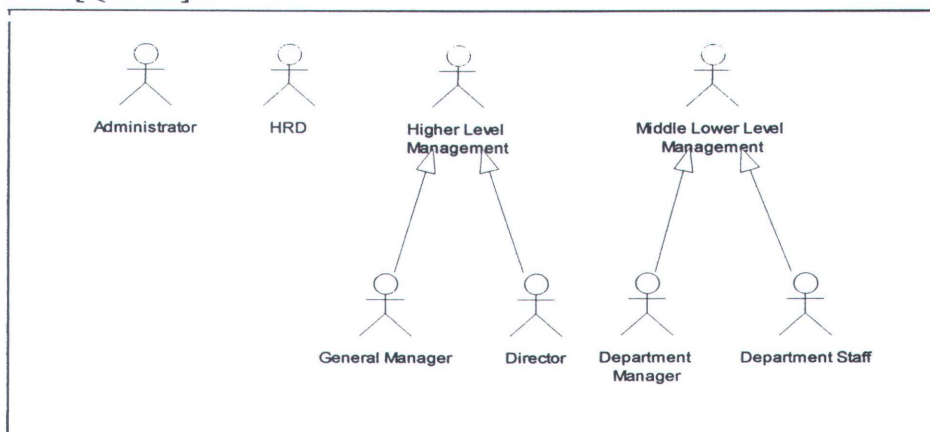
Perancangan proses perangkat lunak ini dilakukan dengan menggunakan notasi UML (*Unified Modelling Language*) yang dimiliki oleh perangkat lunak Rational Rose Enterprise Edition 2002. Rancangan perangkat lunak tersebut berupa diagram *usecase*.

4.2.1. Analisis Usecase View

Tahap ini mempunyai sebuah tujuan, yaitu menggambarkan tingkah laku sistem dengan fungsi-fungsi yang akan disediakan olehnya. Langkah pertama adalah mengidentifikasi aktor yang akan terlibat, yang kedua adalah mendefinisikan *usecase* untuk tiap-tiap aktor dan yang terakhir adalah *activity diagram* untuk *usecase* yang dirasa membutuhkan penjelasan lebih detail.

4.2.2.1. Aktor

Aktor tidak termasuk bagian dari sistem. Aktor merepresentasikan orang atau sesuatu obyek yang berinteraksi dengan sistem. Aktor dapat hanya menerima informasi, memberi informasi atau menerima dan memberi informasi kepada sistem [QUA98].



Gambar 4.2 Aktor-Aktor yang terlibat dalam sistem

Sebagaimana terlihat pada gambar 4.2, pada sistem ini terdapat empat buah aktor yang terlibat, yaitu *Administrator*, *Human Resources Department*, *High Level Management* dan *Middle-Lower Level Management*.

Setiap aktor memiliki penjelasan dan hak akses pada sistem yang berbeda. Berikut ini adalah tabel 4.11 yang berisi daftar, definisi dan hak akses (*role*) setiap aktor yang akan terlibat dalam sistem.

Tabel 4.11 Aktor dan definisi tugasnya dalam sistem

No	Nama Aktor	Definisi	Hak Akses
1	<i>Administrator</i>	Orang yang melakukan <i>maintenance</i> teknis dan membantu operasional sistem.	Semua jenis akses (<i>role</i>) pengguna, yaitu: <ul style="list-style-type: none"> ▪ Manage pengguna dan <i>role</i> ▪ Manage departemen ▪ Input dan update KPI ▪ Setting periode ▪ Input dan update bobot ▪ Input target KPI periodik ▪ Input pencapaian KPI periodik ▪ Melihat hasil pengukuran kinerja
2	<i>Human Resources Department (HRD)</i>	Orang yang merupakan karyawan dari <i>Human Resources Department</i> yang menangani Pengukuran Kinerja. Bertugas untuk <i>maintenance</i> operasional sistem.	Pada hak akses (<i>role</i>) HRD, yaitu: <ul style="list-style-type: none"> ▪ Manage departemen ▪ Input dan update KPI ▪ Setting periode ▪ Input dan update bobot ▪ Melihat hasil pengukuran kinerja
3	<i>High Level Management (HLM)</i>	Orang yang memiliki kedudukan sebagai <i>director</i> dari sebuah direktorat atau <i>General Manager</i> dari setiap divisi.	Pada hak akses (<i>role</i>) HLM, yaitu: <ul style="list-style-type: none"> ▪ Input target KPI periodik ▪ <i>Accountability</i> ▪ Melihat hasil pengukuran kinerja

No	Nama Aktor	Definisi	Hak Akses
4	Middle-Lower Level Management (MLLM)	Orang yang memiliki kedudukan sebagai <i>manager</i> atau sebagai <i>department-staff</i> pada setiap departemen.	Pada hak akses (<i>role</i>) MLLM, yaitu: <ul style="list-style-type: none">▪ Input pencapaian KPI periodik▪ Melihat hasil pengukuran kinerja

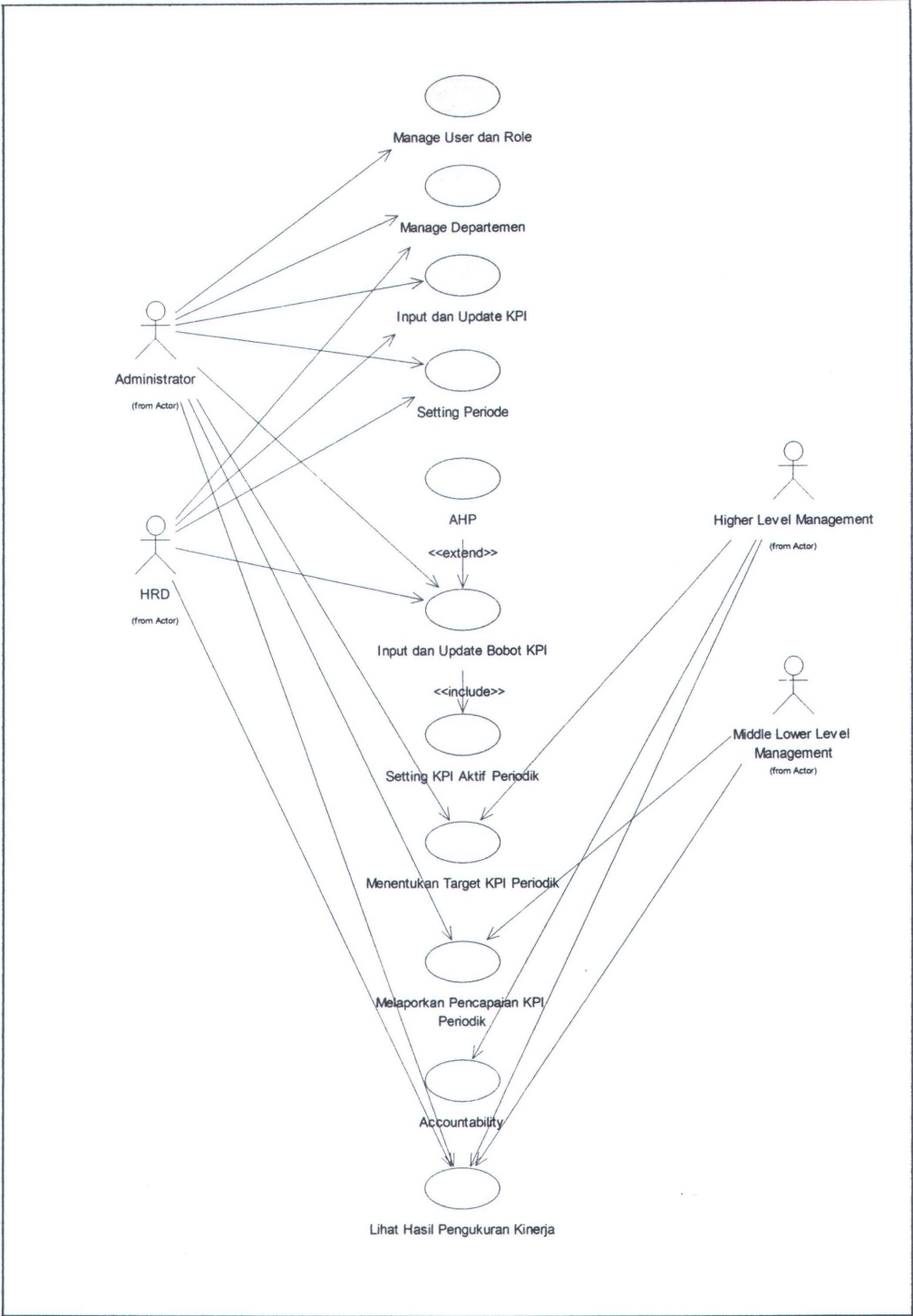
4.2.2.2. Usecase

Model *usecase* adalah dialog antara seorang aktor dengan sistem. Sebuah *usecase* adalah urutan transaksi-transaksi yang dilakukan oleh sebuah sistem yang menyediakan hasil terukur untuk sebagian aktor [QUA98]. *Usecase-usecase* yang ada pada perangkat lunak dijelaskan pada tabel 4.12 berikut.

Tabel 4.12 Daftar Usecase

No	Nama proses	Keterangan
1	Manage pengguna dan <i>role</i>	Proses ini dibutuhkan untuk menambah, mengedit dan menghapus pengguna dan jenis hak akses setiap pengguna.
2	Manage departemen	Proses ini dibutuhkan untuk mengelola data departemen.
3	Input dan update KPI	Proses ini dibutuhkan untuk input dan update KPI.
4	Setting periode	Proses ini dibutuhkan untuk menentukan jangka waktu atau periode pengukuran kinerja.
5	AHP	Proses ini dibutuhkan untuk melakukan pembobotan KPI dengan metode AHP. <i>Usecase</i> bersifat <i>extend</i> .
6	Input dan update bobot KPI	Proses ini dibutuhkan untuk input dan update bobot KPI.
7	Setting KPI aktif periodik	Proses ini dibutuhkan untuk menentukan KPI-KPI yang aktif pada suatu periode tertentu. <i>Usecase</i> bersifat <i>include</i> .
8	Menentukan target KPI periodik	Proses ini dibutuhkan untuk menentukan target KPI periodik.
9	Melaporkan pencapaian KPI periodik	Proses ini dibutuhkan untuk memasukkan pencapaian KPI periodik.
10	<i>Accountability</i>	Proses ini dibutuhkan untuk HLLM melakukan Acc suatu pelaporan.
11	Lihat hasil pengukuran kinerja	Proses ini dibutuhkan untuk melihat hasil pengukuran kinerja. Berupa data dalam bentuk tabel dan grafik. Terdapat dua macam bentuk, yaitu setiap periode atau setiap bulan periode.

Usecase-usecase yang terlibat dalam sistem dapat digambarkan pada *Rational Rose* seperti pada gambar 4.3 berikut. Dan untuk penjelasan lebih terperinci dari setiap *usecase* dibahas pada subbab 4.2.2 pada subbab desain *logical view*.



Gambar 4.3 UseCase-UseCase dalam sistem

4.2.2. Desain Logical View

Setelah membuat *usecase view*, langkah selanjutnya adalah membuat *logical view*. Pada tahap ini akan dibuat realisasi dari *usecase* dalam *usecase realization* dan membuat *activity diagram* untuk menggambarkan aktifitas-aktifitas yang ada dalam menyusun aplikasi Sistem Pengukuran Kinerja.

4.2.2.1. *Usecase Realization*

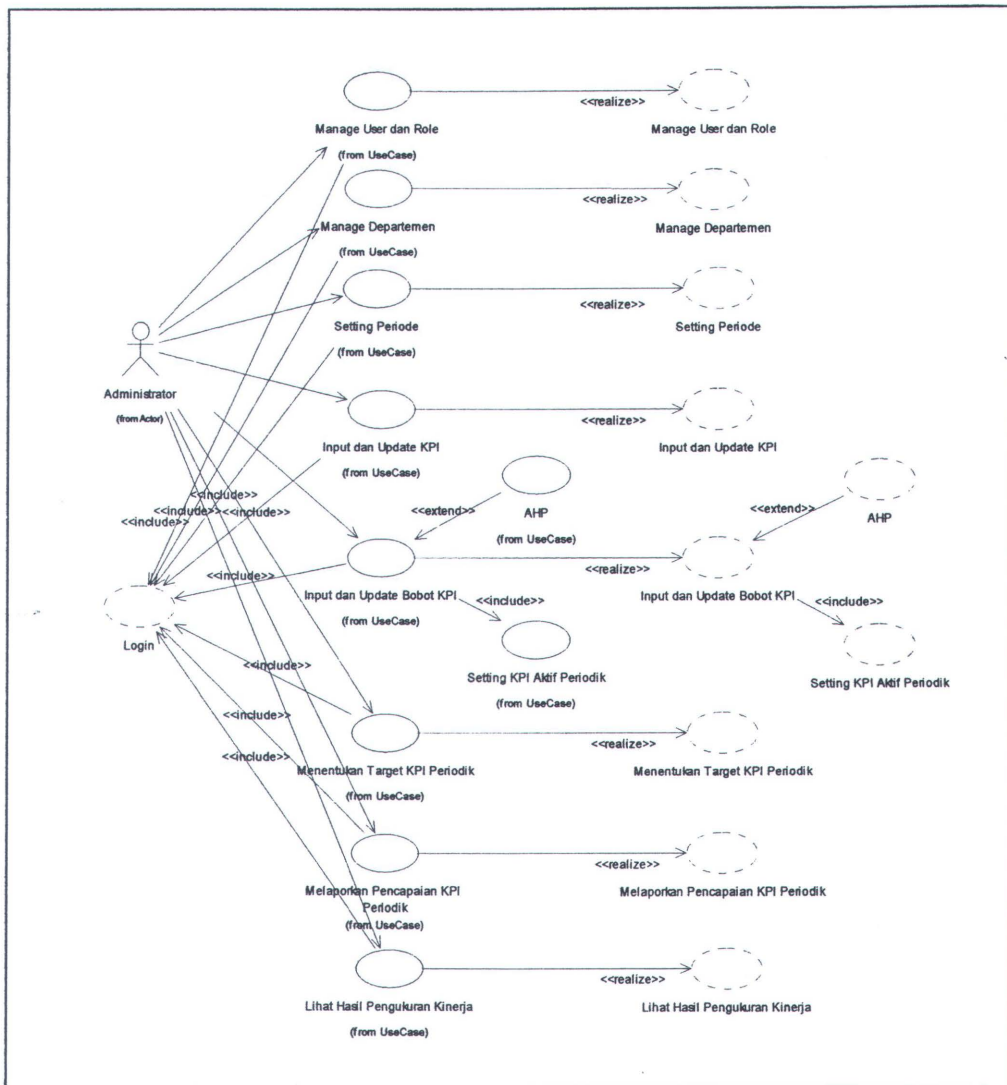
Usecase Realization dibuat untuk menggambarkan menjadi lebih logis dari model-model *usecase* yang telah didesain. *Usecase Realization* sebagai realisasi dari model-model *Usecase* yang telah dibuat ini akan diklasifikasikan berdasarkan Aktor-Aktor yang terlibat dalam sistem.

Dalam sistem pengukuran kinerja ini diklasifikasikan menjadi empat jenis, sebagai berikut:

4.2.2.1.1. *Usecase Realization dari Aktor Administrator*

Setiap *usecase* yang dimiliki oleh aktor administrator direalisasikan ke dalam satu buah *usecase realization*. Aktor administrator ini memiliki delapan buah *usecase* dan direalisasikan menjadi sembilan *usecase realization*, dengan penambahan *usecase realization login* sebagai *autentifikasi user* aktor administrator.

Pada aktor administrator ini juga terdapat satu buah *usecase* yang bersifat *extend* dan juga satu buah *usecase* yang bersifat *include*. Kedua *usecase* yang *extend* dan *include* ini juga direalisasikan menjadi masing-masing *usecase realization* yang berbeda. Untuk lebih jelasnya, dapat dilihat pada gambar 4.4.



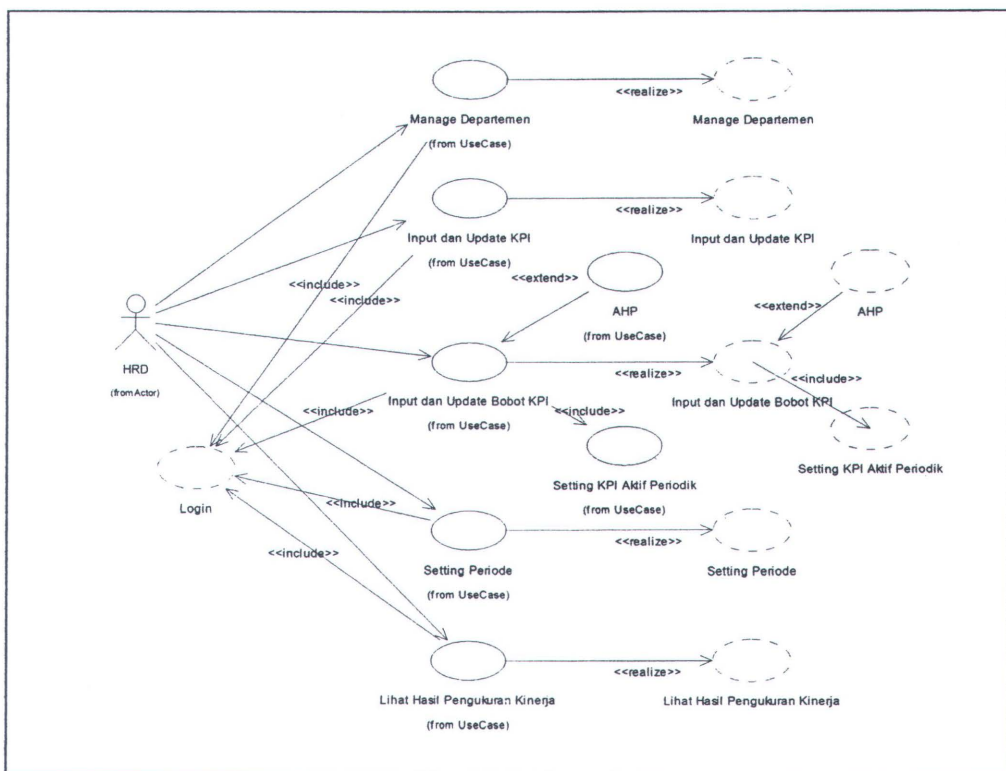
Gambar 4.4 Usecase Realization Administrator

4.2.2.1.2. Usecase Realization dari Aktor HRD

Sebagaimana *usecase realization* pada aktor administrator, maka pada *usecase realization* aktor HRD ini, setiap *usecase* yang dimiliki oleh aktor HRD direalisasikan ke dalam satu buah *usecase realization*. Aktor HRD ini memiliki lima buah *usecase* dan direalisasikan menjadi enam *usecase realization*, dengan penambahan *usecase realization* login sebagai *autentifikasi user* aktor HRD.

Pada aktor HRD ini juga terdapat satu buah *usecase* yang bersifat *extend* dan juga satu buah *usecase* yang bersifat *include*. Kedua *usecase* yang *extend* dan *include* ini juga direalisasikan menjadi masing-masing *usecase realization* yang berbeda.

Sesuai dengan judul Tugas Akhir ini yang menggunakan metode AHP (*Analytical Hierarchy Process*), maka *usecase* AHP direalisasikan menjadi *usecase realization* AHP yang bersifat *extend* pada *usecase* *input dan update bobot KPI* yang dimiliki oleh aktor HRD. Untuk lebih jelasnya, dapat dilihat pada gambar 4.5 berikut.

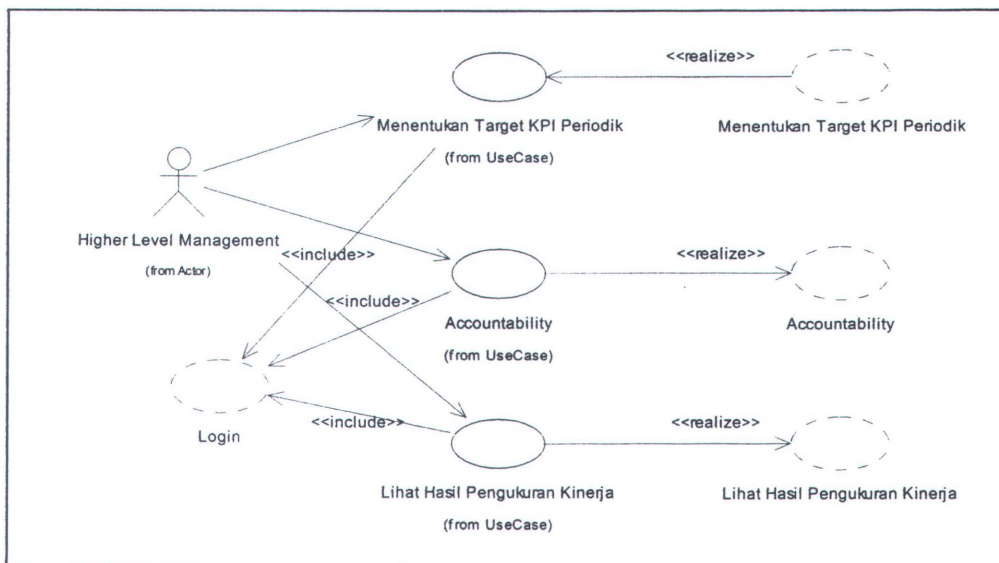


Gambar 4.5 Usecase Realization HRD

4.2.2.1.3. Usecase Realization dari Aktor HLM

Usecase realization dari aktor HLM (*Higher Level Management*) yang mempunyai turunan aktor *Director* dan *General Manager* merupakan realisasi dari aktifitas sistem yang dilakukan oleh kedua turunan aktor tersebut. Dalam hal ini, mereka mempunyai tiga buah *usecase realization* yang merupakan realisasi langsung dari *usecase – usecase* nya, ditambah satu buah *usecase login* sebagai autentifikasi *user* aktor HLM.

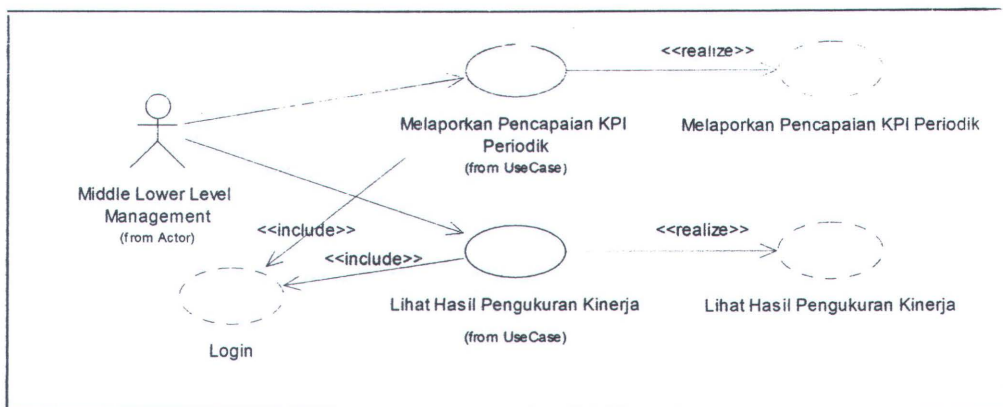
Pada *usecase* HLM ini ada satu buah *usecase* yang tidak dimiliki oleh aktor-aktor yang lain, yaitu *usecase accountability*. *Usecase accountability* ini direalisasikan menjadi *usecase realization accountability*. Untuk lebih jelasnya, dapat dilihat pada gambar 4.6 berikut.



Gambar 4.6 Usecase Realization HLM

4.2.2.1.4. Usecase Realization dari Aktor MLLM

Sebagaimana dijelaskan pada subbab 4.2.2.1. bahwa aktor MLLM (*Middle Lower Level Management*) ini mempunyai turunan aktor *Manager* dan *Department Staff*. *Usecase realization* dari aktor MLLM merupakan realisasi dari aktifitas sistem yang dilakukan oleh kedua turunan aktor tersebut. Dalam hal ini, mereka mempunyai dua buah *usecase realization* yang merupakan realisasi langsung dari *usecase – usecase* nya, ditambah satu buah *usecase login* sebagai autentifikasi *user* aktor MLLM. Untuk lebih jelasnya, dapat dilihat pada gambar 4.7 berikut.



Gambar 4.7 Usecase Realization MLLM

4.2.2.2. Activity Diagram dan Sequence Diagram

Pada pembuatan logical view, setelah pembuatan *usecase realization* dari tiap-tiap *usecase*, selanjutnya membuat activity diagram. Activity diagram memodelkan alur kerja (*workflow*) sebuah proses dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat dimodelkan sebuah alur kerja dari suatu aktivitas ke aktivitas lainnya atau dari satu aktivitas kedalam keadaan sesaat (*state*). Seringkali bermanfaat bila dibuat

sebuah activity diagram terlebih dahulu dalam memodelkan sebuah proses untuk membantu dalam pemahaman proses secara keseluruhan. Activity diagram juga sangat berguna ketika menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *usecase* berinteraksi.

Sequence diagram menggambarkan urutan proses antar *class* dalam melakukan suatu *usecase*. Sequence diagram melibatkan aktor dan *class-class* yang terlibat pada *usecase*, seperti *class boundary*, *entity* dan *class control*.

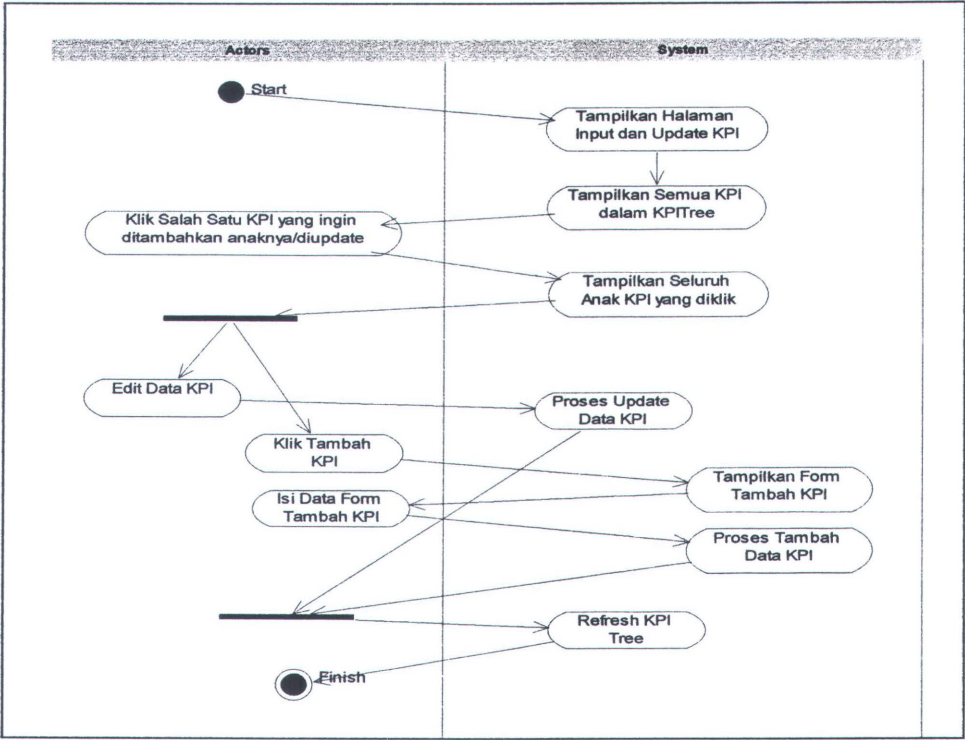
4.2.2.2.1. *Usecase Input dan Update KPI*

Agar sistem pengukuran kinerja dapat dipergunakan maka memerlukan adanya input data-data KPI yang nantinya akan digunakan sebagai indikator pengukuran kinerja.

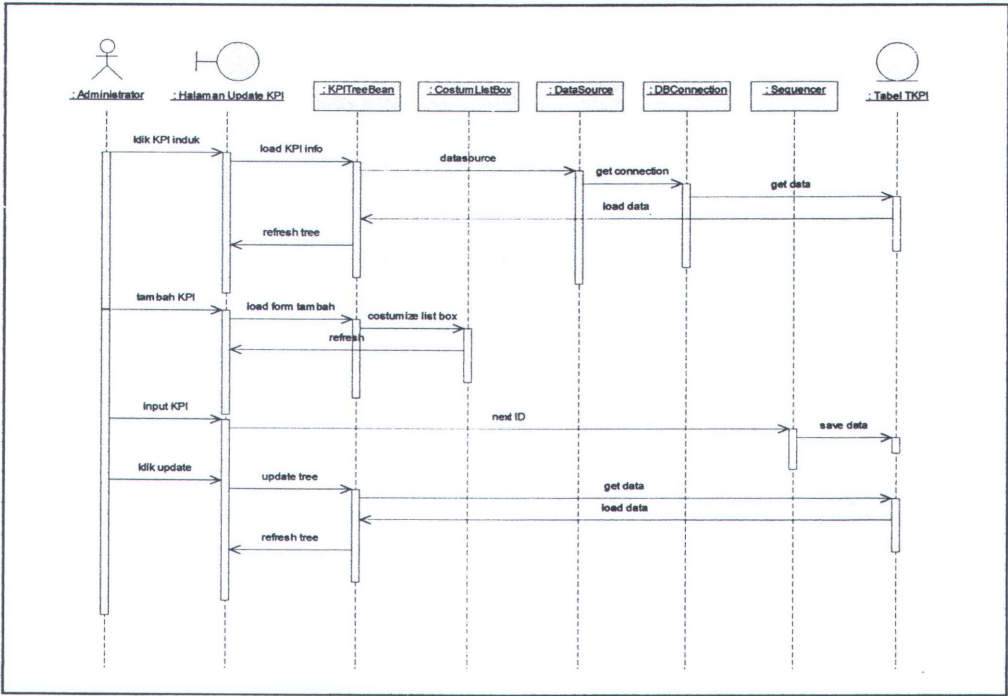
Cara menginputkan dan meng-update KPI disajikan dalam bentuk diagram pohon (*tree*). Tujuannya agar *user* dapat lebih mudah dalam menggambarkan hirarki tiap-tiap KPI dan mempermudah dalam mengetahui induk (*parent*) dan anak (*child*) dari setiap KPI terhadap KPI lain. Selengkapnya dapat dilihat pada activity diagram pada gambar 4.8 dan sequence diagram pada gambar 4.9 berikut.

4.2.2.2.2. *Usecase Setting Periode*

Sistem pengukuran kinerja ini berjalan pada setiap periode pengukuran. Setiap periode dapat memiliki jangka waktu tiga bulan, empat bulan, enam bulan atau lainnya. Oleh karena itu, diperlukan *usecase* yang mengatur dan mengeset periode pengukuran kinerja.



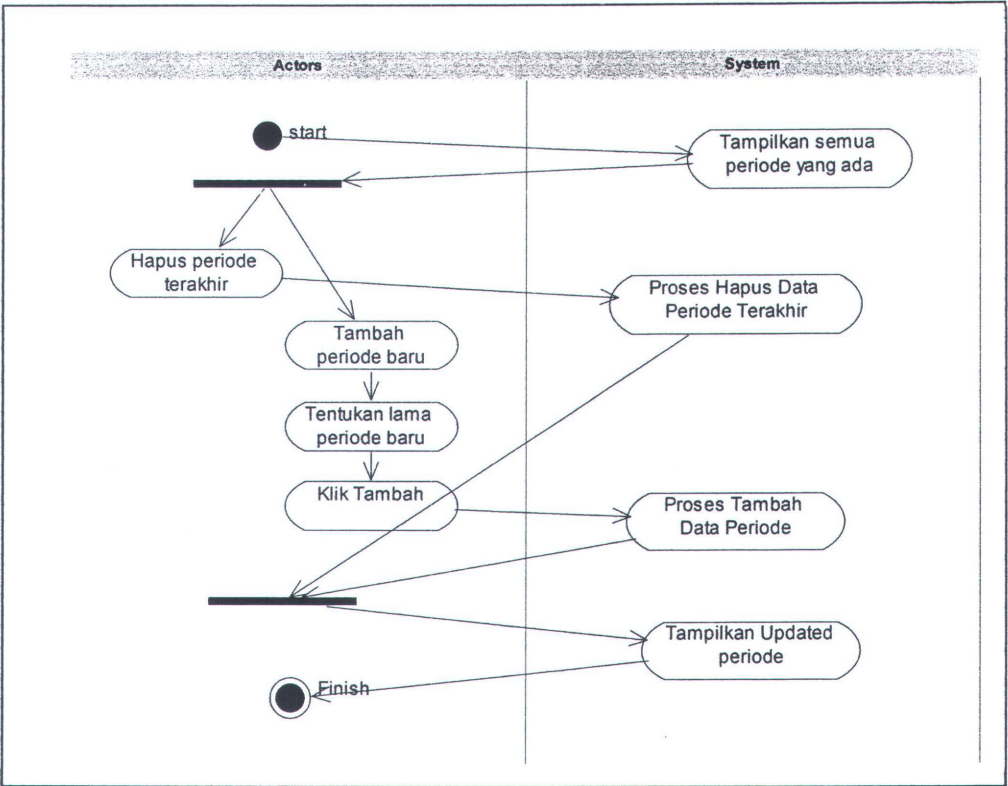
Gambar 4.8 Activity Diagram Input dan Update KPI



Gambar 4.9 Sequence Diagram Input dan Update KPI

Pada *usecase* ini dapat menambahkan periode baru secara otomatis hanya dengan mengeset lama bulannya saja. Untuk aktivitas menghapus periode dilakukan dengan menghapus periode yang terakhir.

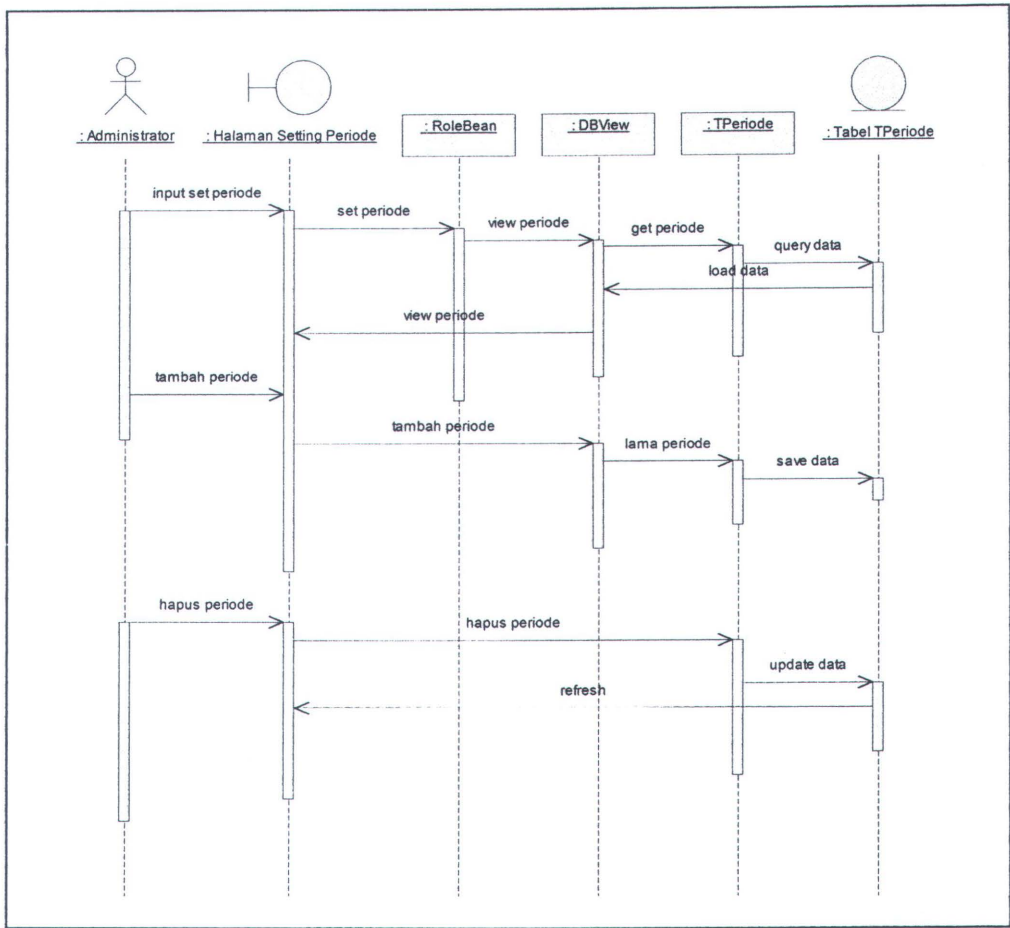
Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.10 dan sequence diagram pada gambar 4.11 berikut.



Gambar 4.10 Activity Diagram Setting Periode

4.2.2.2.3. *Usecase Input dan Update Bobot KPI*

Input dan update bobot KPI diperlukan untuk menentukan bobot setiap KPI yang dipakai dalam pengukuran kinerja. Input dan update bobot KPI ini dilakukan pada setiap awal periode pengukuran.

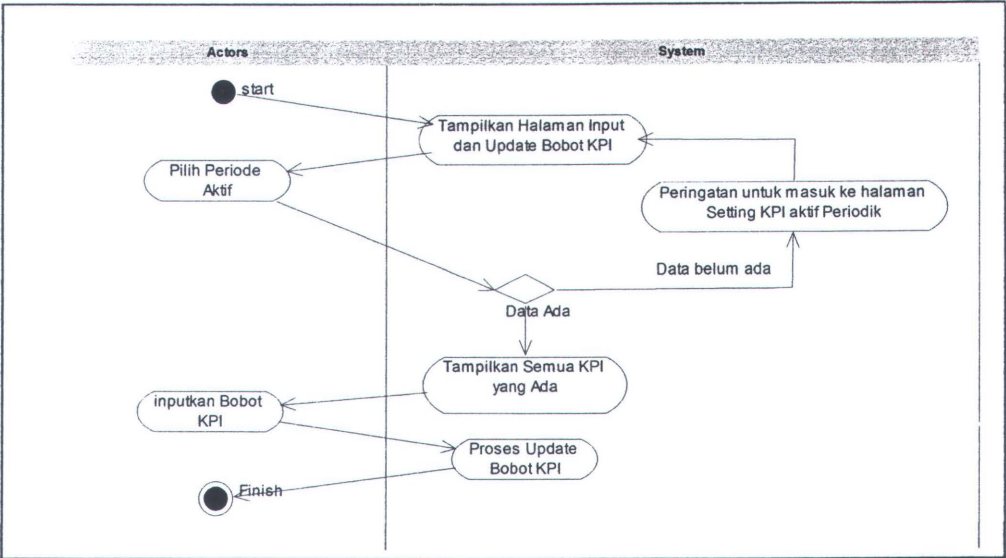


Gambar 4.11 Sequence Diagram Setting Periode

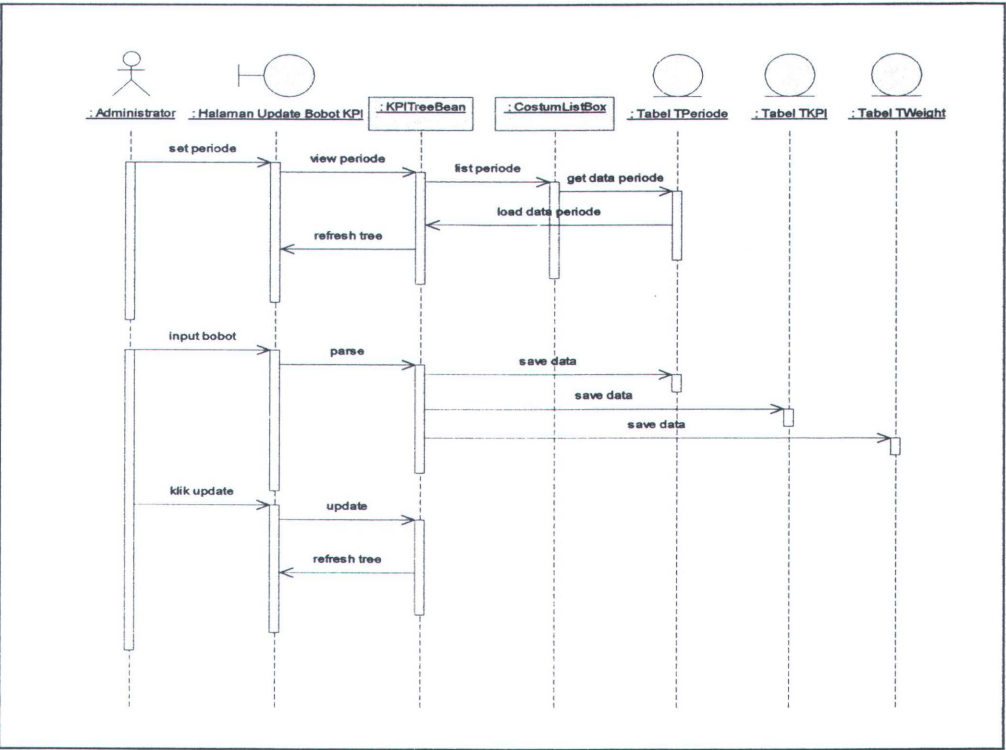
Usecase input dan update KPI ini memiliki satu buah usecase yang extend, yaitu usecase AHP. Maksudnya, ketika usecase ini dilakukan maka dapat saja untuk melakukan usecase AHP ini. Tetapi dapat juga tidak. Jika dari pihak HRD sebagai aktor yang menginputkan dan mengupdate bobot KPI ini merasa perlu menggunakan AHP maka dapat digunakan. Begitu pula sebaliknya.

Usecase input dan update bobot KPI ini juga include terhadap usecase setting KPI aktif periodik. Artinya, ketika usecase ini dijalankan maka harus dahulu melakukan usecase setting KPI aktif periodik.

Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.12 dan sequence diagram pada gambar 4.13 berikut.



Gambar 4.12 Activity Diagram Input dan Update Botot KPI

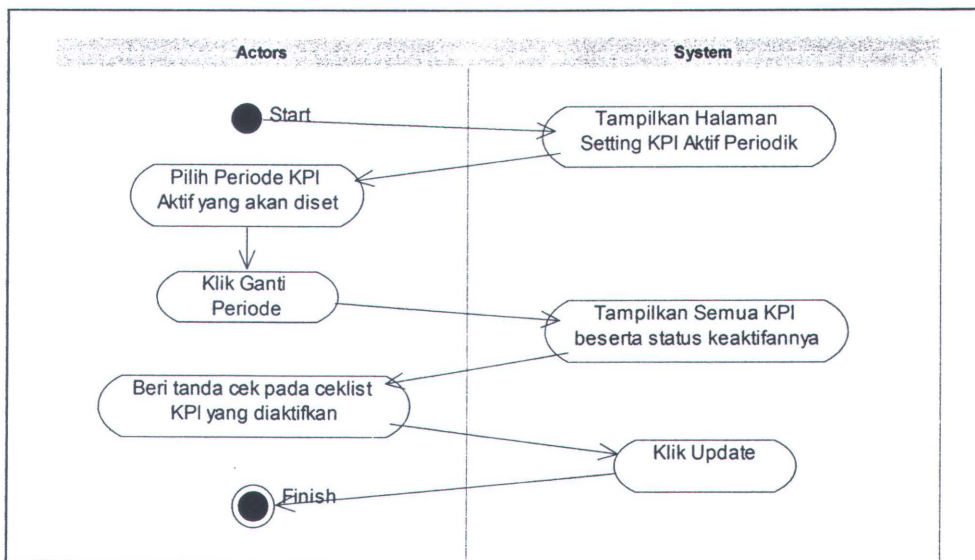


Gambar 4.13 Sequence Diagram Input dan Update Botot KPI

4.2.2.2.4. Usecase Setting KPI Aktif Periodik

Sebagaimana dijelaskan sebelumnya bahwa *usecase* setting KPI Aktif periodik ini merupakan *usecase* yang include terhadap *usecase* input dan update bobot KPI. *Usecase* ini diperlukan untuk menentukan KPI mana saja yang dipakai (di set aktif) pada suatu periode pengukuran kinerja.

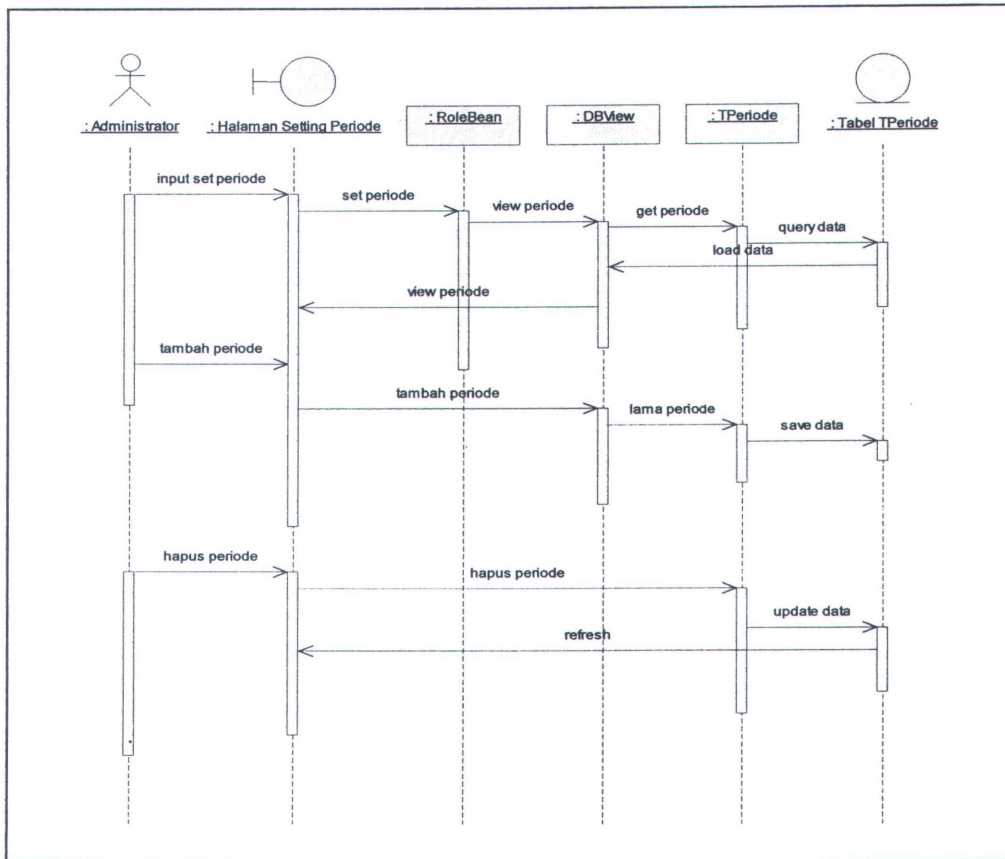
Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.14 dan sequence diagram pada gambar 4.15 berikut.



Gambar 4.14 Activity Diagram Setting KPI Aktif Periodik

4.2.2.2.5. Usecase AHP

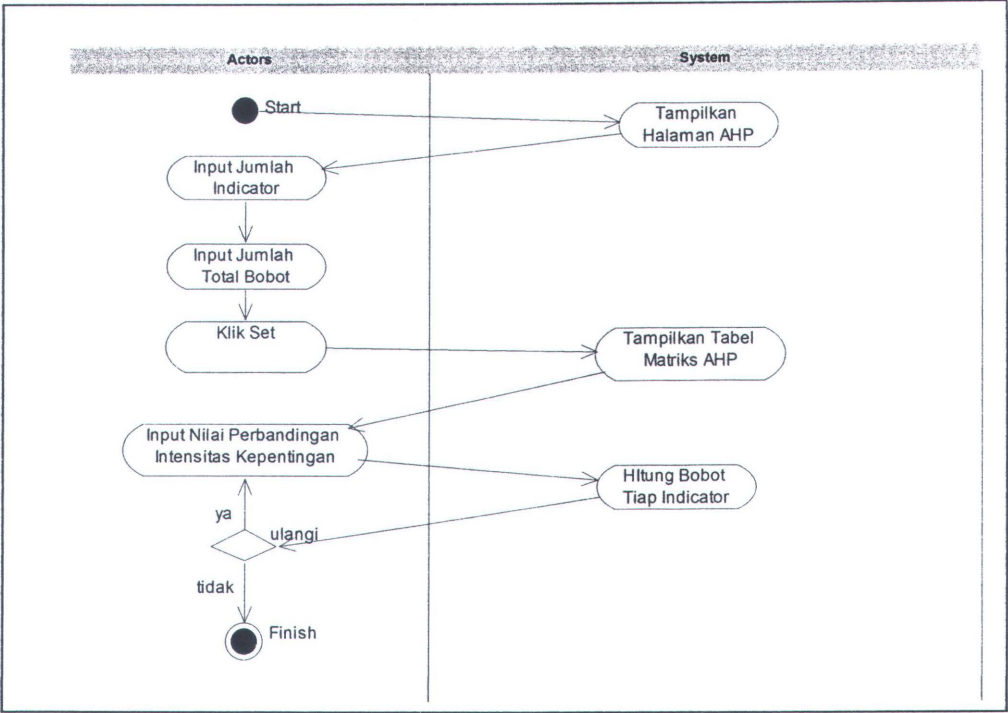
Usecase AHP ini merupakan metode yang digunakan dalam melakukan pembobotan perhitungan kinerja. *Usecase* ini extend terhadap *usecase* input dan update bobot KPI.



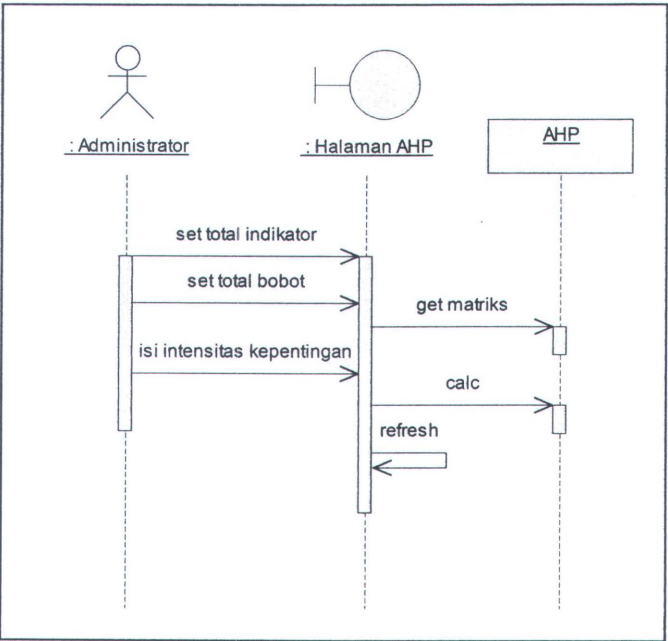
Gambar 4.15 Sequence Diagram Setting KPI Aktif Periodik

Usecase AHP ini adalah *usecase* yang digunakan untuk menghitung bobot suatu KPI. Metode AHP ini berupa tabel matriks bujursangkar sebagai metode untuk perhitungannya. Caranya adalah dengan menentukan berapa jumlah indikator yang dipakai pada suatu level hirarki. Kemudian dengan menginputkan nilai intensitas kepentingan antara satu indikator terhadap indikator yang berhubungan.

Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.16 dan sequence diagram pada gambar 4.17 berikut.



Gambar 4.16 Activity Diagram AHP

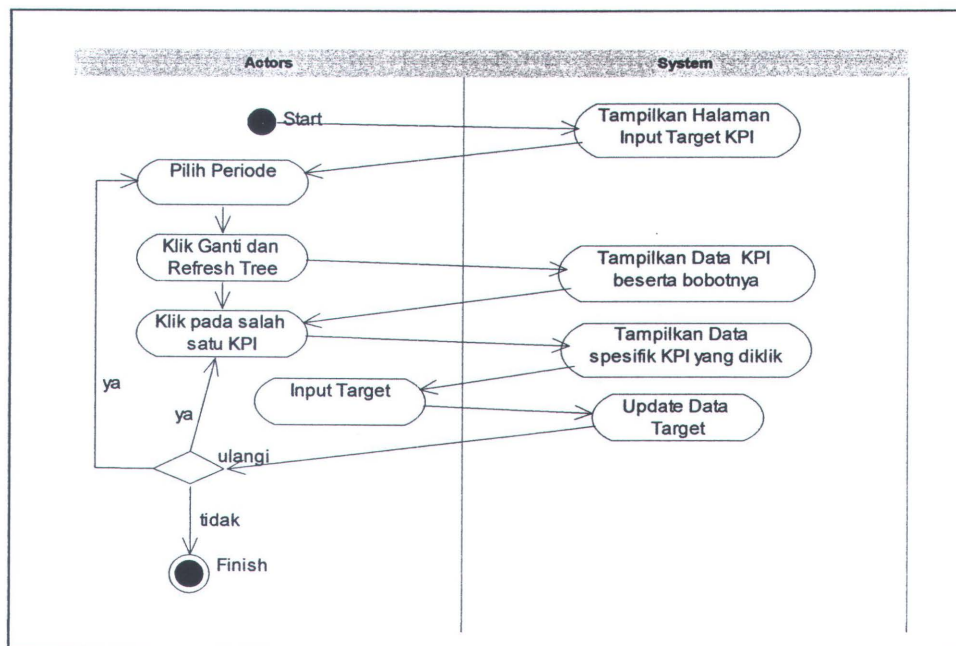


Gambar 4.17 Sequence Diagram AHP

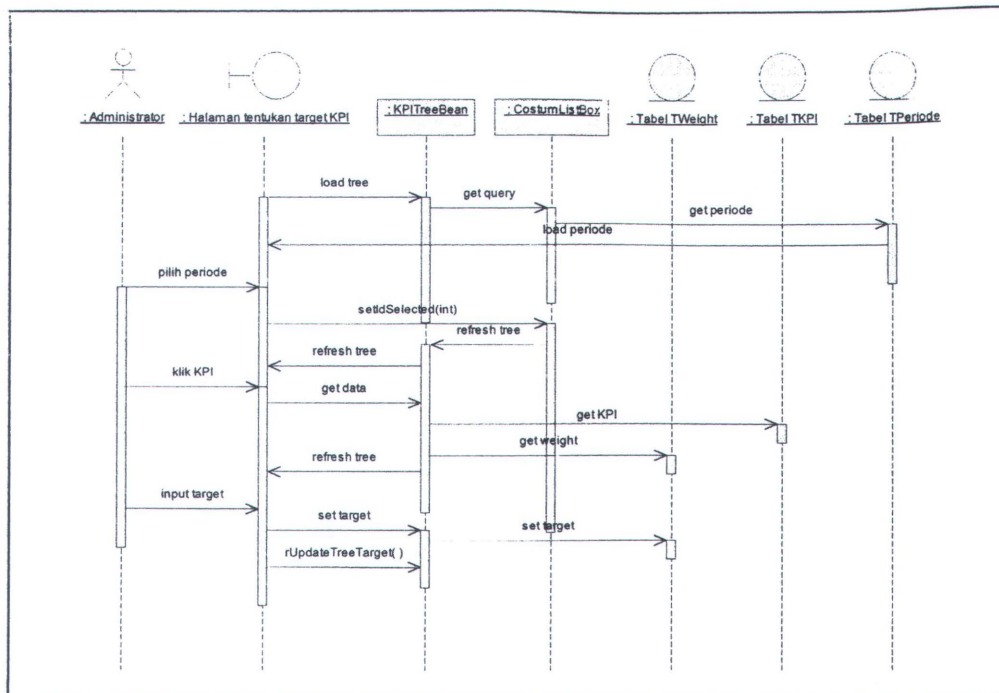
4.2.2.2.6. Usecase Menentukan Target KPI Periodik

Pada setiap awal pengukuran kinerja, pihak *Higher Level Management* yang terdiri dari Direktur dan General Manager menentukan target yang harus dicapai oleh tiap-tiap indikator pada setiap periode. Menentukan target KPI periodik ini dilakukan setelah KPI yang ada diaktifkan dan setelah KPI-KPI tersebut memiliki bobot pengukuran kinerja. Kedua *precondition* ini dilakukan oleh aktor HRD.

Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.18 dan sequence diagram pada gambar 4.19 berikut.



Gambar 4.18 Activity Diagram Menentukan Target KPI Periodik



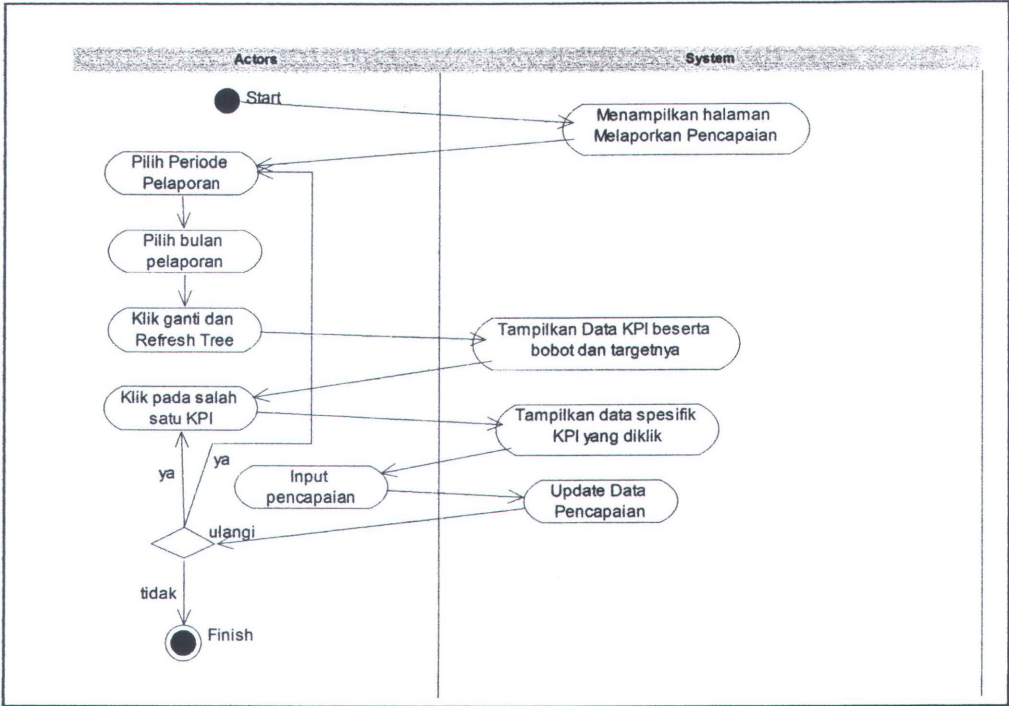
Gambar 4.19 Sequence Diagram Menentukan Target KPI Periodik

4.2.2.2.7. Usecase Melaporkan Pencapaian KPI Periodik

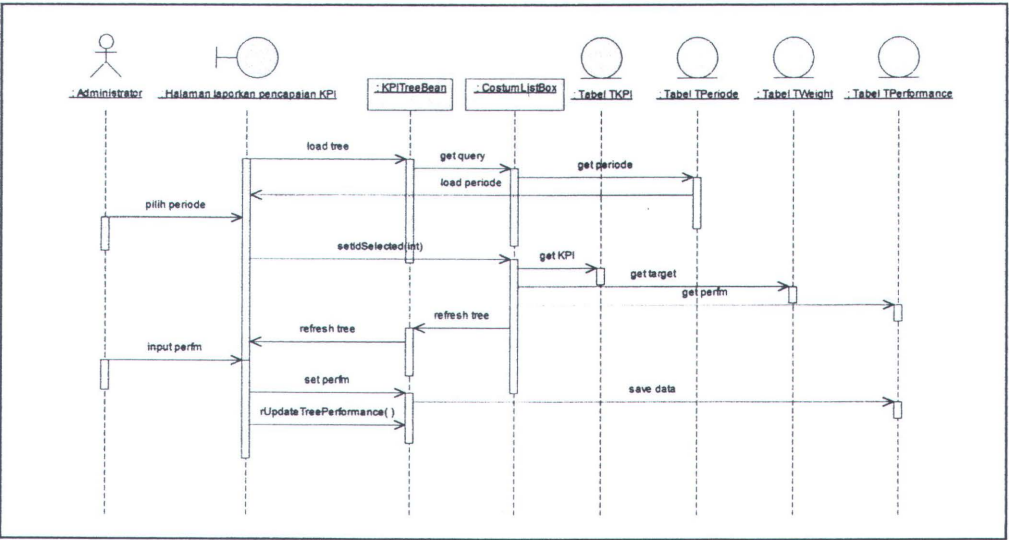
Usecase melaporkan pencapaian KPI periodik dilakukan oleh aktor MLLM. Pelaporan dapat dilakukan secara bertahap pada setiap bulannya, atau langsung diakumulasi selama beberapa bulan pada akhir periode pengukuran kinerja.

Precondition untuk *usecase* ini adalah *user* harus melewati autentifikasi *user* dengan login sesuai dengan jenis hak akses dan asal departemen *user*.

Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.20 dan sequence diagram pada gambar 4.21 berikut.



Gambar 4.20 Activity Diagram Melaporkan Pencapaian KPI Periodik



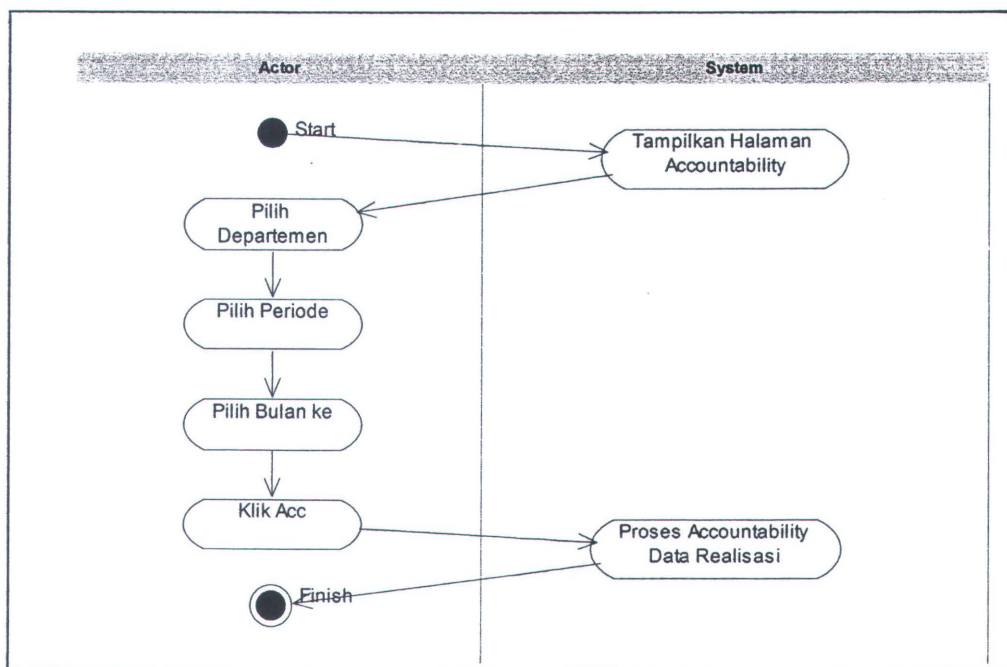
Gambar 4.21 Sequence Diagram Melaporkan Pencapaian KPI Periodik

4.2.2.2.8. Usecase Accountability

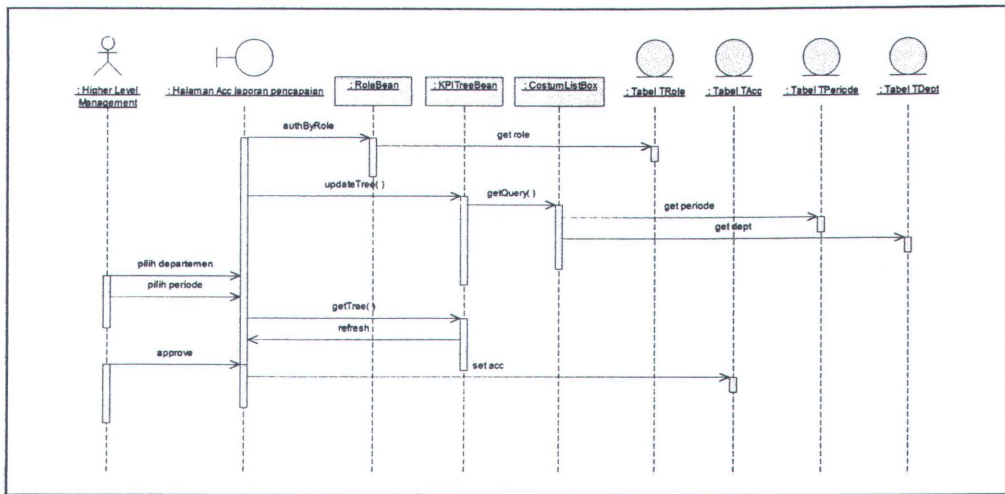
Usecase Accountability dilakukan oleh aktor HLM, yaitu pihak manajemen yang lebih tinggi dari departemen yang bersangkutan. *Usecase* ini perlu sebagai konfirmasi kevalidan informasi pelaporan hasil pencapaian kinerja sebuah KPI.

Precondition yang harus ada adalah selain autentifikasi login dari *user* yang berhak melakukan *accountability*, juga perlu adanya pengisian data yang sudah lengkap pada bobot, target dan pencapaian masing-masing KPI.

Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.22 dan sequence diagram pada gambar 4.23 berikut.



Gambar 4.22 Activity Diagram Accountability

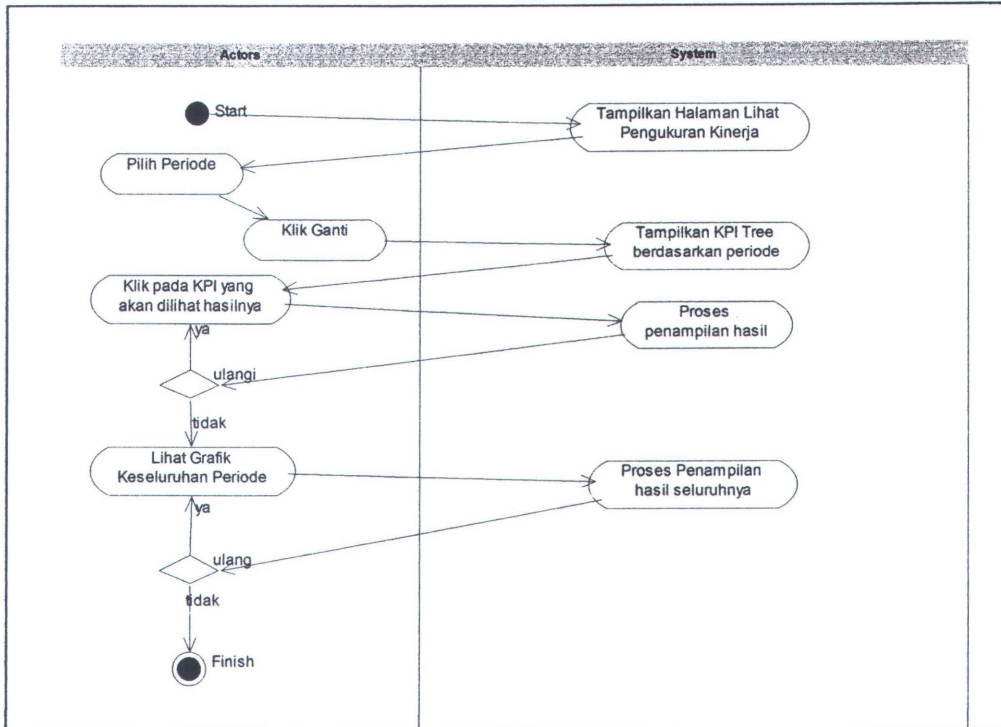


Gambar 4.23 Sequence Diagram Accountability

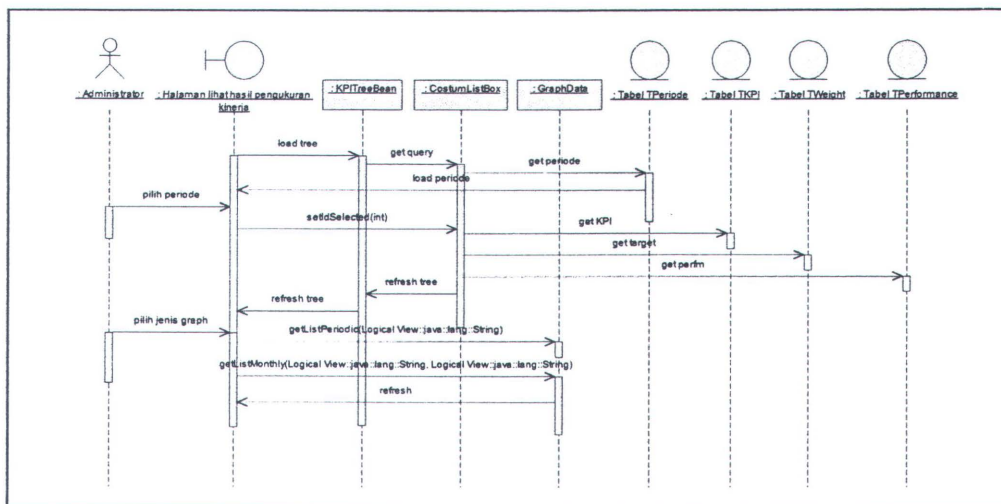
4.2.2.2.9. Usecase Lihat Hasil Pengukuran Kinerja

Setelah semua data diinputkan maka akan dilihat hasil kinerja dari suatu perusahaan dengan berdasarkan pada *framework Balance score card*. Hasilnya ditampilkan dalam bentuk angka-angka kuantitatif dan berupa grafik batang. Grafik menampilkan pencapaian kinerja per KPI per bulan per periode.

Cara menghitung hasil pengukuran kinerja ini dengan membandingkan antara target yang ditetapkan oleh HLM pada awal periode dengan hasil pencapaian yang dilaporkan oleh pihak MLLM. Selengkapnya dapat dilihat pada gambar activity diagram pada gambar 4.24 dan sequence diagram pada gambar 4.25 berikut.



Gambar 4.24 Activity Diagram Lihat Hasil Pengukuran Kinerja



Gambar 4.25 Sequence Diagram Lihat Hasil Pengukuran Kinerja

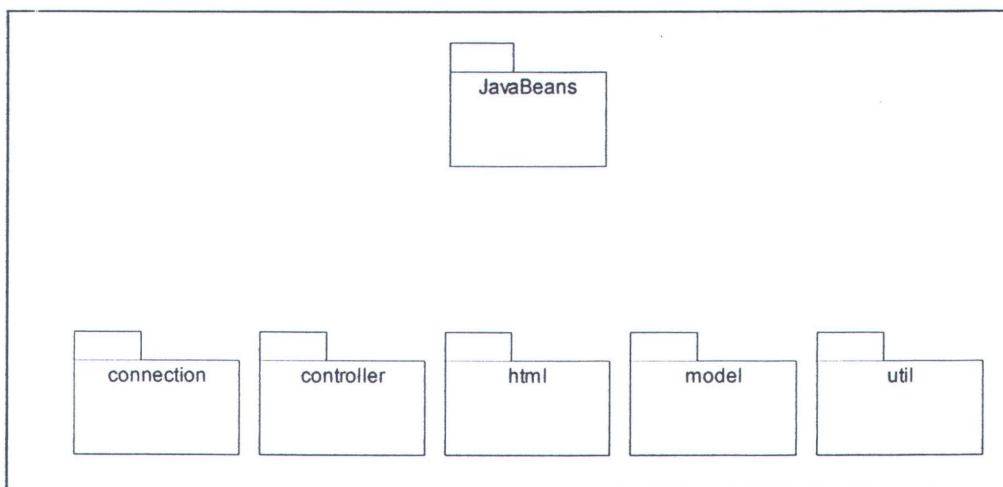
4.2.3. Desain Class *Package*

Dalam pembuatan aplikasi pengukuran kinerja ini banyak menggunakan class-class yang dikelompokkan kedalam beberapa *packages*. Ada beberapa macam *packages* yang digunakan dalam aplikasi. Untuk lebih jelasnya akan dijelaskan sebagai berikut:

4.2.3.1. Package JavaBeans

Package JavaBeans ini berisi semua class-class yang digunakan sebagai komponen JavaBeans. Sebagaimana dijelaskan dalam bab 3 mengenai JavaServer Pages dan JavaBeans, bahwa penggunaan JavaServer Pages akan lebih dinamis bila menggunakan komponen JavaBeans.

Dalam *Package* JavaBeans ini sendiri berisi *package-package* lagi yang dikelompokkan menjadi lima buah *package*, yaitu *package model*, *package html*, *package controller*, *package util* dan *package connection*. Agar lebih jelas dapat digambarkan pada gambar 4.26 berikut.



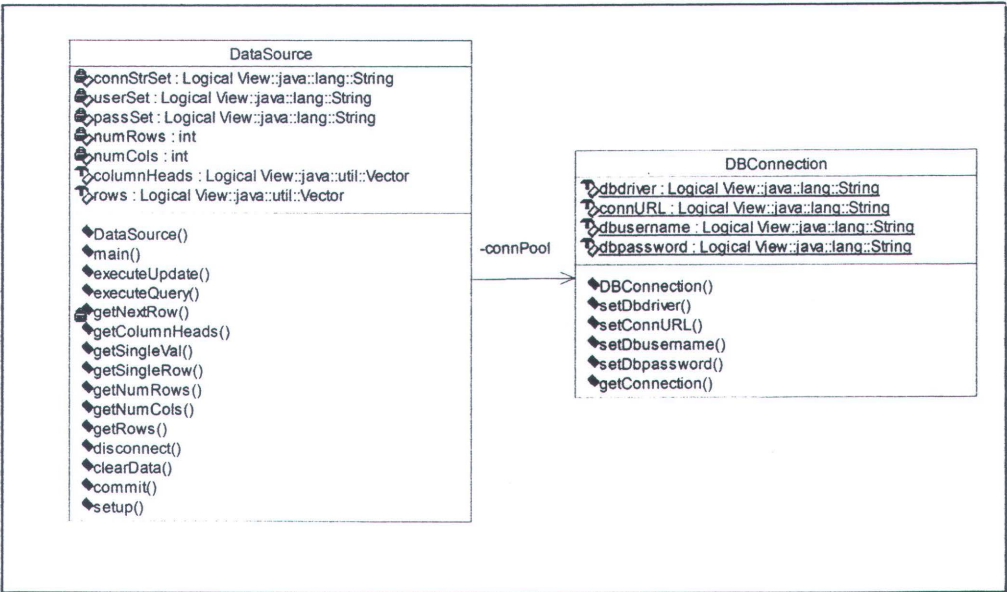
Gambar 4.26 Package JavaBeans

4.2.3.2. Package Connection

Package connection berisi class-class yang mengatur koneksi antara sistem dengan database. Dalam *package connection* terdapat dua buah class yaitu *DBConnection.java* dengan *DataSource.java*.

Untuk *DBConnection.java* berfungsi untuk melakukan koneksi dengan database, yaitu meliputi pengesetan database driver, URL, database *user* dan database *password*. Dalam aplikasi ini digunakan database microsoft sqlserver.

Untuk *DataSource.java* mengatur mengenai pengaturan resultset untuk query database. Selengkapnya dapat dilihat pada gambar 4.27 berikut:



Gambar 4.27 Class-class pada Package Connection

4.2.3.3. Package Controller

Package controller berisi class-class yang berfungsi sebagai kontrol dan *business logic*. Sebagaimana dibahas pada bab 3 mengenai paradigma desain MVC (*model-view-controller*), maka *package* controller ini adalah bagian dari desain MVC tersebut.

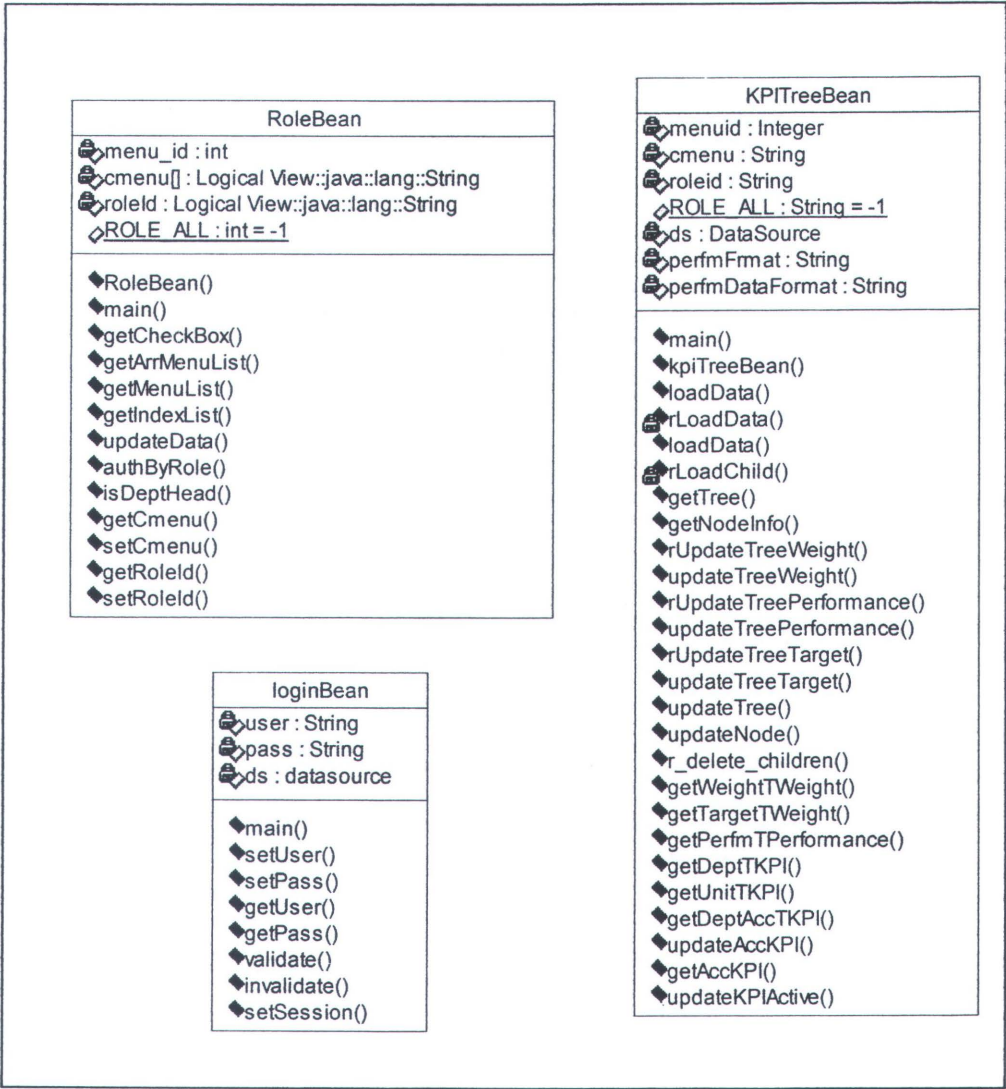
Package controller berisi tiga buah class, antara lain *RoleBean.java*, *loginBean.java* dan *KPITreeBean.java*.

Untuk *RoleBean.java* mengatur tentang pengaturan menu bagi setiap jenis *role*. Sebagai input adalah jenis *role*. Untuk *loginBean.java* mengurus tentang *username* dan *password* user serta membuat *session* bagi user.

Dan untuk *KPITreeBean.java* adalah JavaBeans yang mengatur tentang penampilan data-data dalam bentuk tree. Selengkapnya dapat dilihat pada gambar 4.28 berikut.

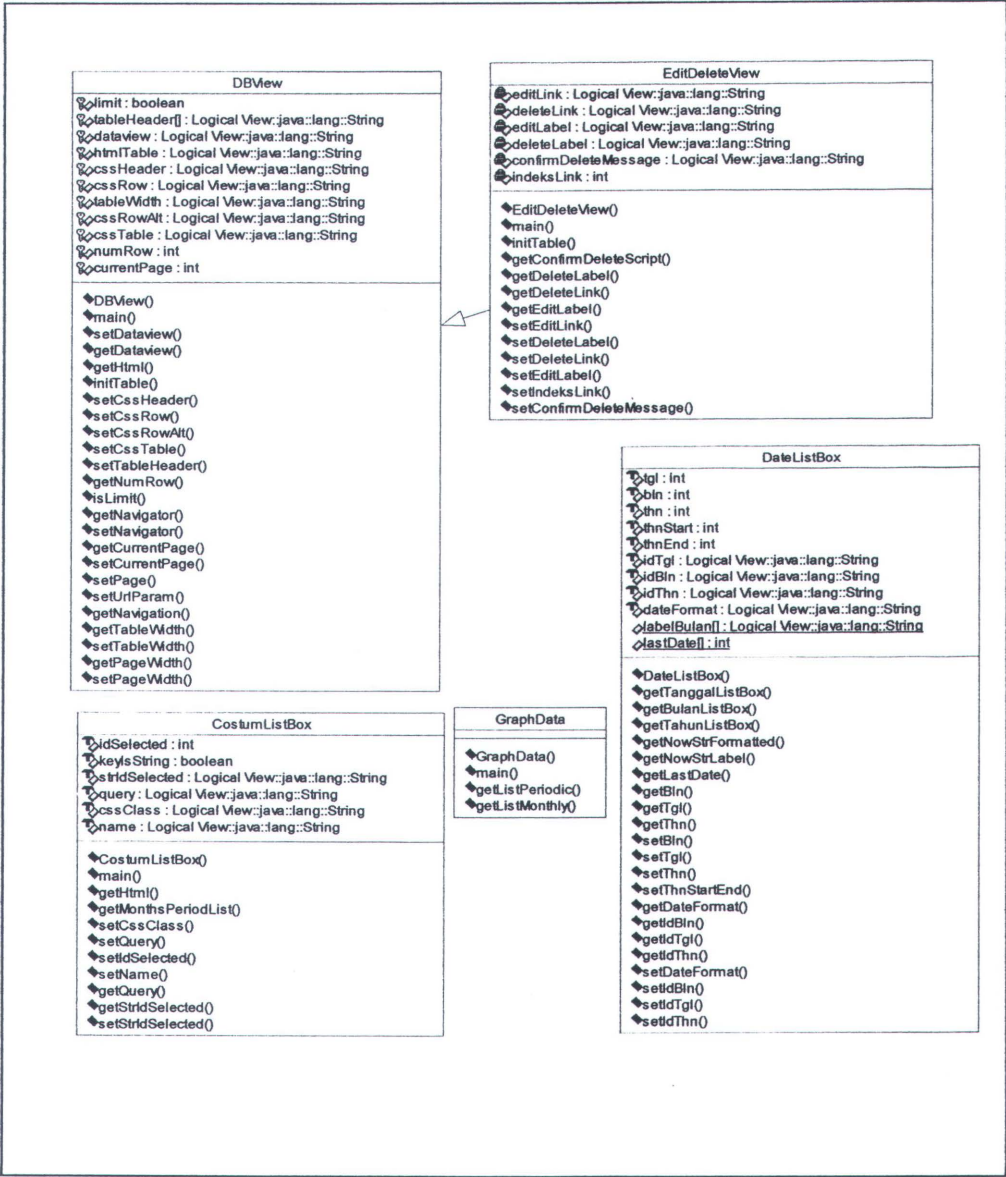
4.2.3.4. Package Html

Package Html ini sebagai komponen yang berperan sebagai view dalam desain MVC. *Package* Html terdiri dari lima buah class, antara lain *CustomListBox.java*, *DateListBox.java*, *DBView.java*, *EditDeleteView.java* dan *GraphData.java*.



Gambar 4.28 Class-class pada Package Controller

Sesuai dengan nama class-nya, *CustomListBox.java* berfungsi untuk mengatur listbox agar dapat dinamis, *DateListBox.java* berfungsi untuk list box yang menampilkan tanggal, dan seterusnya.

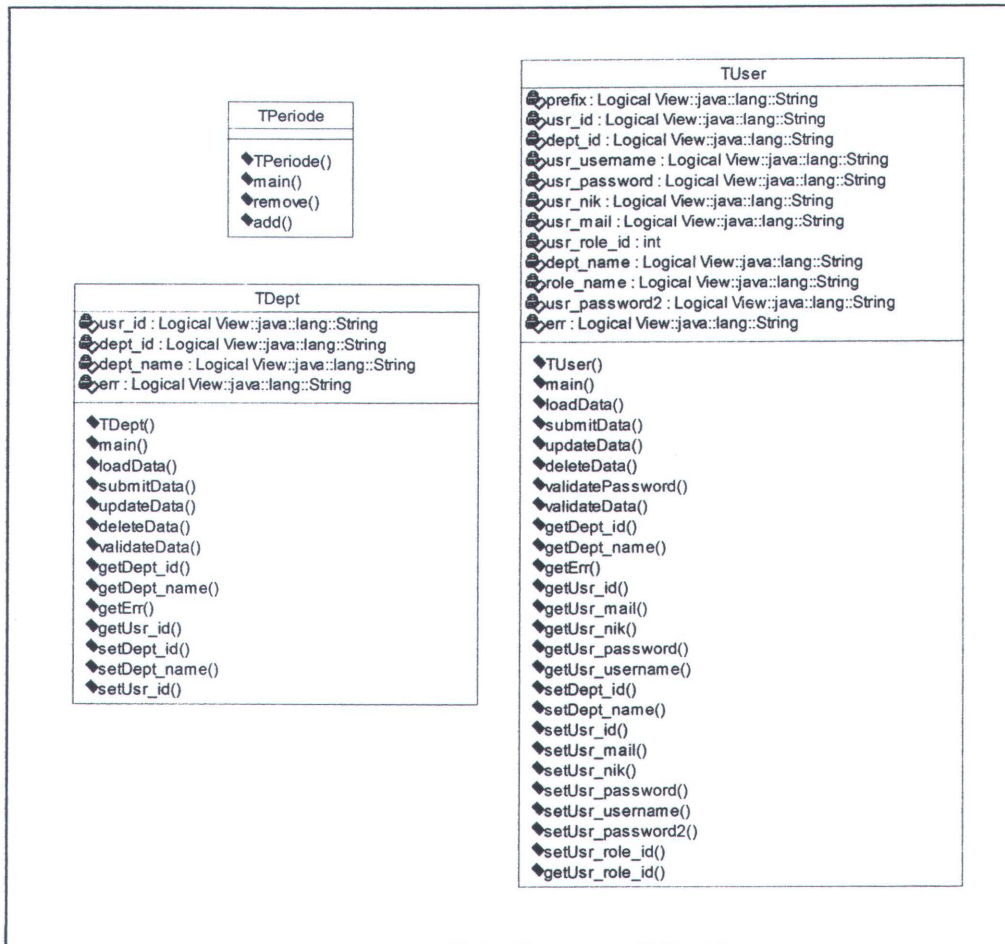


Gambar 4.29 Class-class pada Package Html

4.2.3.5. Package Model

Package model berisi tentang class-class yang mengatur setiap manipulasi data tabel pada database. Dalam aplikasi ini ada tiga buah tabel yang diatur oleh desain MVC yaitu Tuser, Tperiode dan Tdept. Class-class pada *package* model ini mengatur setiap *get* dan *set* setiap atribut dari setiap tabel.

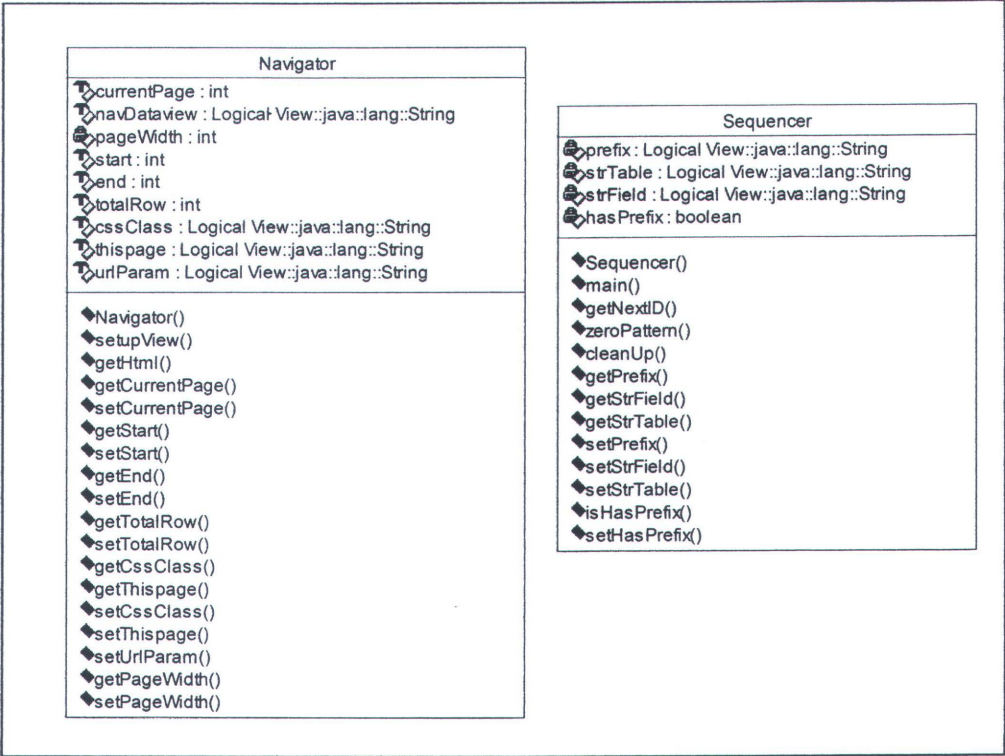
Selengkapnya dapat dilihat pada gambar 4.30 berikut:



Gambar 4.30 Class-class pada Package Model

a. **Package Util**

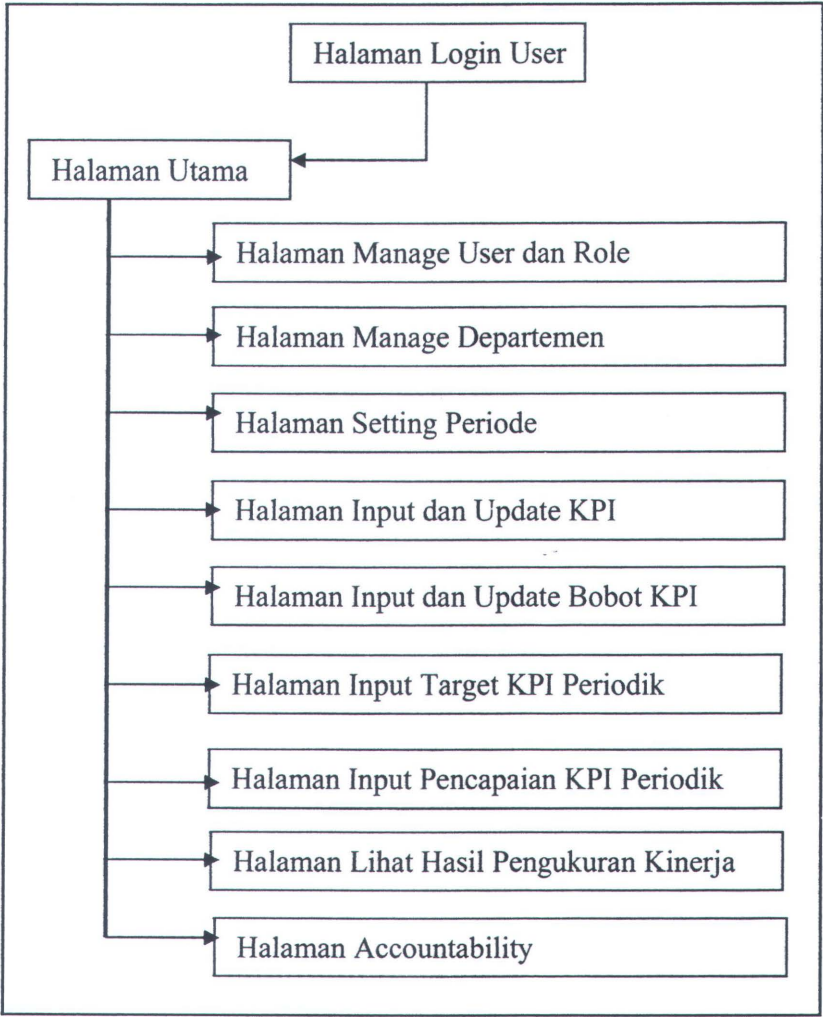
Package util berisi perangkat komponen tambahan dalam mengatur tampilan. Ada dua buah class, yaitu *navigator.java* dan *sequencer.java*. Untuk *navigator.java* mengatur tampilan tabel, *paging*, dan beberapa fungsi untuk navigasi. Selengkapnya dapat dilihat pada gambar 4.31.



Gambar 4.31 Class-class pada Package Util

4.3. DESAIN ANTARMUKA

Antarmuka merupakan komponen yang menghubungkan perangkat lunak dengan pengguna. Untuk Desain antarmuka ini sebenarnya telah dibahas pada subbab 4.2.2.1 tentang *usecase realization*. Pada *usecase realization* merupakan realisasi dari setiap *usecase* yang sudeah berinteraksi langsung dengan aktor. Namun, pada subbab ini merekap secara lebih umum dari desain antarmuka tersebut. Gambar 4.32 menunjukkan desain antarmuka perangkat lunak.



Gambar 4.32 Desain Antarmuka

*Hidup ini bagai hanya berteduh sejenak..
Sekejap saja..!!*

BAB V

IMPLEMENTASI PERANGKAT LUNAK

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



BAB V

IMPLEMENTASI PERANGKAT LUNAK

Bab ini menjelaskan tentang tahap implementasi perangkat lunak. Tahap implementasi perangkat lunak dilakukan setelah tahap desain selesai. Meskipun pada pelaksanaannya, akan ada proses dimana desain yang telah dibuat perlu diubah kembali untuk menyesuaikan dengan proses penulisan program. Class-class yang telah didefinisikan pada tahap desain akan diimplementasikan pada bahasa pemrograman Java. Dan untuk desain *usecase-usecase* pada bagian antarmuka akan diimplementasikan dalam bahasa JavaServer Pages.

5.1. IMPLEMENTASI DATA

Implementasi data meliputi implementasi desain *data model diagram* menjadi tabel. Pada implementasi data ini menggunakan Microsoft SQL Server 2000 sebagai piranti basis datanya.

a. Implementasi Tabel TUser

Implementasi tabel TUser pada gambar 5.1 berikut.

```
create table TUSER (  
    USR_ID          numeric(5)          not null,  
    DEPT_ID         numeric(3)          not null,  
    ROLE_ID         numeric(3)          not null,  
    USR_USERNAME    varchar(11)         null,  
    USR_PASSWORD    varchar(11)         null,  
    USR_NIK         varchar(9)          null,  
    USR_MAIL        varchar(35)         null,  
    constraint PK_TUSER primary key (USR_ID)  
) go
```

Gambar 5.1 Implementasi Tabel TUser

b. Implementasi Tabel TDept

Implementasi tabel TDept pada gambar 5.2 berikut.

```
create table TDEPT (
    DEPT_ID          numeric(3)          not null,
    USR_ID           numeric(5)          null,
    DEPT_NAME        varchar(35)         null,
    constraint PK_TDEPT primary key (DEPT_ID)
) go
```

Gambar 5.2 Implementasi Tabel TDept

c. Implementasi Tabel TPeriode

Implementasi tabel TPeriode pada gambar 5.3 berikut.

```
create table TPERIODE (
    PER_ID           numeric(3)          not null,
    PER_START        datetime            null,
    PER_FINAL        datetime            null,
    constraint PK_TPERIODE primary key (PER_ID)
) go
```

Gambar 5.3 Implementasi Tabel TPeriode

d. Implementasi Tabel TKPI

Implementasi tabel TKPI pada gambar 5.4 berikut.

```
create table TKPI (
    KPI_ID           numeric(5)          not null,
    DEPT_ID          numeric(3)          not null,
    TDE_DEPT_ID      numeric(3)          not null,
    KPI_NAME         varchar(40)         null,
    KPI_DESC         varchar(150)        null,
    KPI_UNIT         varchar(20)         null,
    KPI_ROOT         varchar(5)          null,
    KPI_NEGATE       bit                 null,
    constraint PK_TKPI primary key (KPI_ID)
) go
```

Gambar 5.4 Implementasi Tabel TKPI

e. Implementasi Tabel TWeight

Implementasi tabel TWeight pada gambar 5.5 berikut.

```
create table TWEIGHT (
    KPI_ID          numeric(5)          not null,
    PER_ID          numeric(3)          not null,
    TWTARGET        varchar(8)          null,
    TWWEIGHT        varchar(8)          null,
    constraint PK_TWEIGHT primary key  (KPI_ID, PER_ID)
) go
```

Gambar 5.5 Implementasi Tabel TWeight

f. Implementasi Tabel TPerformance

Implementasi tabel TPerformance pada gambar 5.6 berikut.

```
create table TPERFORMANCE (
    KPI_ID          numeric(5)          not null,
    PER_ID          numeric(3)          not null,
    PERFM_MONTH     numeric(2)          not null,
    USR_ID          numeric(5)          not null,
    PERFM_VALUE     varchar(25)         null,
    PERFM_ENTRY     datetime            null,
    constraint PK_TPERFORMANCE primary key  (KPI_ID, PER_ID,
    PERFM_MONTH)
) go
```

Gambar 5.6 Implementasi Tabel TPerformance

g. Implementasi Tabel TAcc

Implementasi tabel TAcc pada gambar 5.7 berikut.

```
create table TACC (
    DEPT_ID        char(7)              not null,
    KPI_ID          numeric(5)          not null,
    PER_ID          numeric(3)          not null,
    PERFM_MONTH     numeric(2)          not null,
    ACC            bit                  null,
) go
```

Gambar 5.7 Implementasi Tabel TAcc

h. Implementasi Tabel TRole

Implementasi tabel TRole pada gambar 5.8 berikut.

```
create table TROLE (
    ROLE_ID          numeric(18)          NOT NULL ,
    ROLE_NAME        varchar(50)          NULL ,
    ROLE_DESC        varchar(50)          NULL ,
    ) constraint PK_TROLE primary key  (ROLE_ID)
) go
```

Gambar 5.8 Implementasi Tabel TRole

i. Implementasi Tabel TMenu

Implementasi tabel TMenu pada gambar 5.9 berikut.

```
create table TMENU (
    MENU_ID          int                  NOT NULL ,
    MENU_NAME        varchar(50)          NULL ,
    MENU_LINK        varchar(50)          NULL ,
    MENU_DESC        varchar(50)          NULL ,
    ) constraint PK_TMENU primary key  (MENU_ID)
) go
```

Gambar 5.9 Implementasi Tabel TMenu

j. Implementasi Tabel TUserRoleMenu

Implementasi tabel TUserRoleMenu pada gambar 5.10 berikut.

```
create table TUSER_ROLE_MENU (
    ROLE_ID          numeric(18)          NOT NULL ,
    MENU_ID          int                  NOT NULL ,
    ) constraint PK_TMENU primary key  (ROLE_ID, MENU_ID)
) go
```

Gambar 5.10 Implementasi Tabel TUserRoleMenu

5.2. IMPLEMENTASI PROSES

Implementasi proses meliputi implementasi dari setiap *usecase* pada bahasa JavaServer Pages yang berperan sebagai proses dan berupa komponen JavaBeans yang juga berperan sebagai proses.

Pada subbab ini, mengingat banyaknya proses yang diimplementasikan, maka hanya akan dibahas beberapa implementasi realisasi *usecase*. Tidak semua implementasi realisasi *usecase* akan dibahas pada buku Tugas Akhir ini.

5.2.1. Implementasi Proses Login

Pada implementasi proses login ini, terdapat komponen JavaBeans *loginBean.java* yang menangani proses login. Selanjutnya, untuk menangani pengecekan hak akses (*role*) dari setiap *user* yang login, terdapat file sebagai JavaBeans implementasi proses.

Kedua komponen JavaBeans ini mengambil koneksi *resultset* pada Beans *DataSource.java* dengan sintaks:

```
import kamal.ta.beans.connection.DataSource;
```

Selengkapnya dapat melihat pada penggalan implementasi *loginBean.java* pada gambar 5.11 dan *roleBean.java* pada gambar 5.12.

5.2.2. Implementasi Proses Manage Departemen

Implementasi proses pada manage departemen ini menggunakan JSP dengan file *dept_edit_act.jsp* yang menangani pengeditan departemen pada gambar 5.13. Juga terdapat *dept_tambah_act.jsp* yang menangani proses penambahan departemen pada gambar 5.14.

```

package kamal.ta.beans.controller;
import java.util.Vector;
import javax.servlet.http.HttpSession;
import kamal.ta.beans.connection.DataSource;
import kamal.ta.beans.model.TUser;

public class loginBean {
    public final int HDEPANADMIN = 1;
    public final int HLOGOUT = 2;
    public final int HGAGALLOGIN = 3;
    HttpSession session;
    String test;
    String user;
    String pass;
    DataSource ds ;

    public void setPass(String string) {
        pass = string;
    }
    public void setUser(String string) {
        user = string;
    }
    public int invalidate()
    {
        if(this.session!=null)
            this.session.invalidate();
        return this.HLOGOUT;
    }
    public int validate()
    {
        int retval=-1;
        int result=-1;
        //... dipotong pada baris ini ...
        return retval;
    }
    public String getPass() {
        return pass;
    }
    public String getUser() {
        return user;
    }
    public String getTest() {
        test = "sipp!";
        return test;
    }
    public void setSession(HttpSession sess){
        this.session = sess;
    }
}

```

Gambar 5.11 Penggalan implementasi loginBean.java.

```

package kamal.ta.beans.controller;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Vector;
import kamal.ta.beans.connection.DataSource;

public class RoleBean {
private int menu_id=0;
private String[] cmenu;
private String roleId = "";
public static final int ROLE_ALL = -1;
private DataSource ds;
public static void main(String[] arg){
    RoleBean b_role = new RoleBean();
    boolean retval = b_role.authByRole("Target","admin");
    System.out.println(retval);
}
public Object[] getCheckBox(int role_id){
    boolean subflag = true;
    Vector fullmenu = this.getMenuList(RoleBean.ROLE_ALL);
    Vector submenu = this.getIndexList(role_id);
    // ... dipotong pada baris ini ...
    return retval.toArray();
}
public String[][] getArrMenuList(int role_id)
{
    Vector v_menu = this.getMenuList(role_id);
    String[][] retval= new String[v_menu.size()][];
    Iterator itr = v_menu.iterator();
    //... dipotong pada baris ini ...
    return retval;
}
public Vector getMenuList(int role_id){
    //... dipotong pada baris ini...
    return retval;
}
public Vector getIndexList(int role_id){
    //... dipotong pada baris ini ...
    return retval;
}
public void updateData()
{ //... dipotong pada baris ini...
    ds.disconnect();
}
public boolean authByRole(String menu_name, String user_name)
{ //... dipotong pada baris ini ...
    return retval;
}
public boolean isDeptHead(String user_name, String dept_id)
{ //... dipotong pada baris ini ...
    return retval;
}
public String[] getCmenu() {
    return cmenu;
}
}
}

```

Gambar 5.12 Penggalan implementasi roleBean.java.

Untuk penggunaan JavaBeans proses pada manage departemen ini menggunakan *roleBean.java* yang sudah dibahas pada subbab 5.2.1 sehingga tidak akan dibahas kembali.

```
%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="dataBean" scope="page"
class="kamal.ta.beans.model.TDept">
  <jsp:setProperty name="dataBean" property="*" />
</jsp:useBean>

<%
String dept_id =request.getParameter("ID");
dataBean.setDept_id( dept_id);
String nextURL;
boolean fl = dataBean.validateData();
if(fl)
{
    dataBean.updateData();
    nextURL = "dept_lihat.jsp?userMsg=suksesNgupdate";
}else{
    nextURL = "dept_edit.jsp?ID="+dept_id+"&errMsg=" +
dataBean.getErr();
}
response.sendRedirect(nextURL);
%>
```

Gambar 5.13 Penggalan implementasi dept_edit_act.jsp

5.2.3. Implementasi Proses Setting Periode

Implementasi proses pada setting periode pada file JSP dengan *periode_act.jsp* pada gambar 5.15. Pada file ini memanggil file JavaBeans *Tperiode.java* sebagai data khusus untuk menangani tabel *TPeriode*.

```
<jsp:useBean id="dataBean" scope="page"
class="kamal.ta.beans.model.TDept">
  <jsp:setProperty name="dataBean" property="*" />
</jsp:useBean>

<%
String nextURL;
boolean fl = dataBean.validateData();
```

Gambar 5.14 Penggalan implementasi dept_tambah_act.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="dataBean" scope="page"
class="kamal.ta.beans.model.TPeriode"/>

<%
String task = request.getParameter("task");
String mon = request.getParameter("mon");
if("hapus".equals(task))
{
dataBean.remove();
}
else if("tambah".equals(task))
{
    int intMon = Integer.parseInt(mon);
    dataBean.add(intMon);
}

response.sendRedirect("periode_main.jsp");
%>

```

Gambar 5.15 Penggalan implementasi periode_act.jsp

5.2.4. Implementasi Proses Input dan Update KPI

Implementasi proses input dan update KPI ini menggunakan komponen *KPITreeBean.java* untuk menangani penampilan tree data-datanya. Begitu juga untuk beberapa implementasi dari beberapa realisasi *usecase* lain.

Mengingat sangat banyaknya *attribute* dan *operation* yang ada pada *KPITreeBean.java* maka tidak akan dibahas pada buku Tugas Akhir ini.

5.2.5. Implementasi Proses pada Input dan Update Bobot KPI

Implementasi proses pada input dan update bobot KPI ini juga menggunakan *KPITreeBean.java* sebagaimana telah dibahas pada subbab 5.2.4 diatas.

5.3. IMPLEMENTASI ANTARMUKA

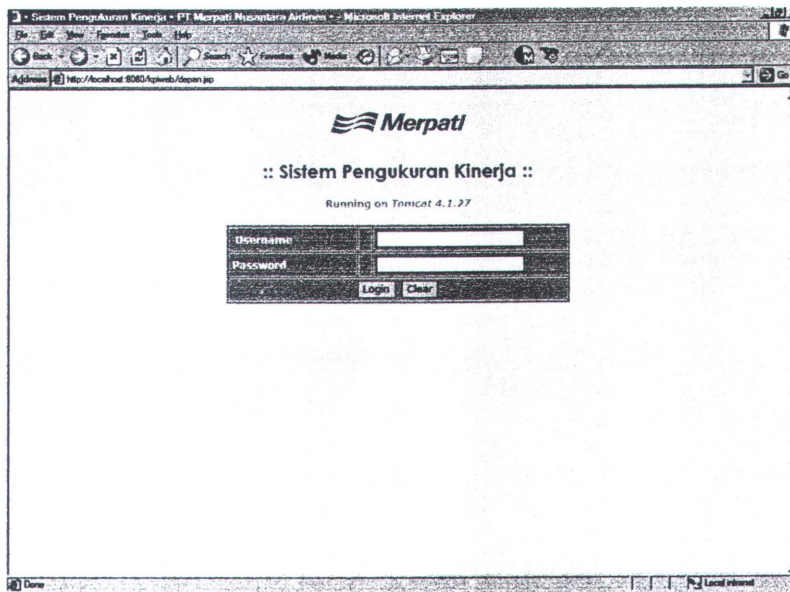
Antarmuka meliputi implementasi dari setiap *usecase* pada bahasa JavaServer Pages. Mengingat jumlah implementasi antarmuka yang banyak maka hanya sebagian antarmuka saja yang ditampilkan pada buku Tugas Akhir ini. Tidak semua antarmuka yang terimplementasi akan dibahas pada buku Tugas Akhir ini. Sebagai contoh akan dibahas lima buah realisasi antarmuka pada implementasi antarmuka ini.

5.3.1. Implementasi Antarmuka Login

Pada *usecase* login, sebagai antarmuka terdapat halaman utama dengan *form* login yaitu *depan.jsp* yang berperan sebagai halaman awal. Setelah *user* memasukkan *username* dan *password*, data input akan dicek oleh *ceklogin.jsp*.

```
<form action="/kpiweb/ceklogin.jsp" method="post">
<input type="hidden" name="action" value="login">
  <table width="45%" border="1" align="center" cellpadding="3"
bordercolor="#999999" bgcolor="#999999">
    <tr valign="middle" bordercolor="#CCCCCC">
      <td width="38%"> &nbsp;<font color="#FFFFFF" size="2"
face="Verdana, Arial, Helvetica, sans-
serif"><strong>Username</strong></font>
      </td>
      <td width="4%">
        <div align="center">:</div></td>
      <td width="58%">
        <input name="user" type="text" id="user" value="admin"
size="25"> </td>
    </tr>
    <tr valign="middle" bordercolor="#CCCCCC">
      <td><font color="#FFFFFF" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><strong>Password</strong></font></td>
      <td align="center">:</td>
      <td>
        <input name="pass" type="password" id="pass" value="admin"
size="25"></td>
    </tr>
    <tr valign="middle" bordercolor="#CCCCCC">
      <td colspan="3" align="center">
        <input name="Cari" type="submit" id="Login"
value="Login">&nbsp;  
        <input name="Clear" type="reset" id="Clear"
value="Clear"></td>
    </tr>
  </table> </form>
```

Gambar 5.16 Implementasi Antarmuka Login



Gambar 5.17 Tampilan Antarmuka Login

5.3.2. Implementasi Antarmuka Manage Departemen

Implementasi antarmuka pada manage departemen dalam bentuk JSP pada file *dept_lihat.jsp* pada gambar 5.18. Pada file *dept_lihat.jsp* ini akan memanggil JavaBeans yang juga berperan sebagai antarmuka yaitu *editDeleteView.java*.

Selanjutnya, ketika *user* akan menambah departemen maka akan memanggil file *dept_tambah.jsp*. Selanjutnya file *dept_tambah.jsp* akan memanggil JavaBeans *CostumListBox.java* sebagai antarmuka juga.

```
<jsp:useBean id="userListBean" scope="page"
class="kamal.ta.beans.html.CostumListBox"/>
<jsp:setProperty name="userListBean" property="query" value="select
usr_id + ' - ' + usr_username , usr_id from TUser"/>
<jsp:setProperty name="userListBean" property="name" value="usr_id"/>

<%@ include file="../script/header.html"%>
<jsp:include page="../script/topmenu.jsp"/>

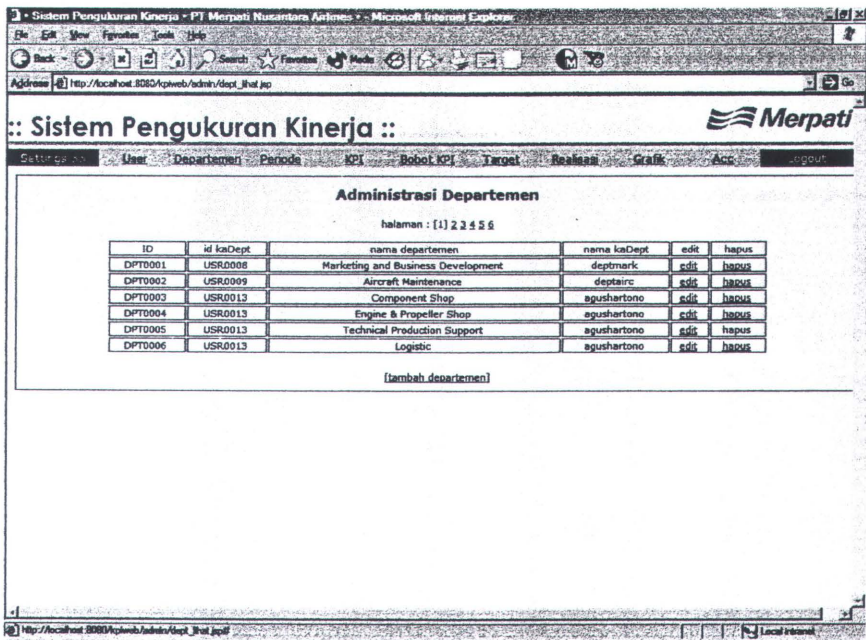
<jsp:getProperty name="userListBean" property="html"/>
```

Gambar 5.18 Implementasi Antarmuka Manage Departemen

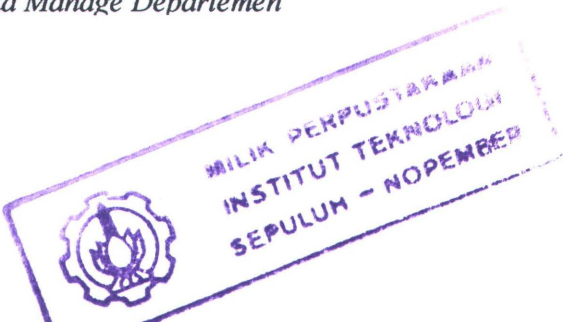
```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="myBean" scope="page"
class="kamal.ta.beans.html.EditDeleteView"/>
<jsp:useBean id="navBean" scope="page"
class="kamal.ta.beans.util.Navigator"/>
<jsp:useBean id="authBean" scope="page"
class="kamal.ta.beans.controller.RoleBean"/>
<%

HttpSession mysession = request.getSession();
String sess_name = (String)mysession.getAttribute("name");
if(sess_name==null || "".equals(sess_name))
{
%>
<jsp:forward page="/depan.jsp?status=timeout"/>
<%
}
if(!authBean.authByRole("Departemen",sess_name))
{
%>
<jsp:forward page="/accessdenied.jsp"/>
<%
}
%>
```

Gambar 5.19 Implementasi Antarmuka Tambah Departemen



Gambar 5.20 Tampilan Antarmuka Manage Departemen



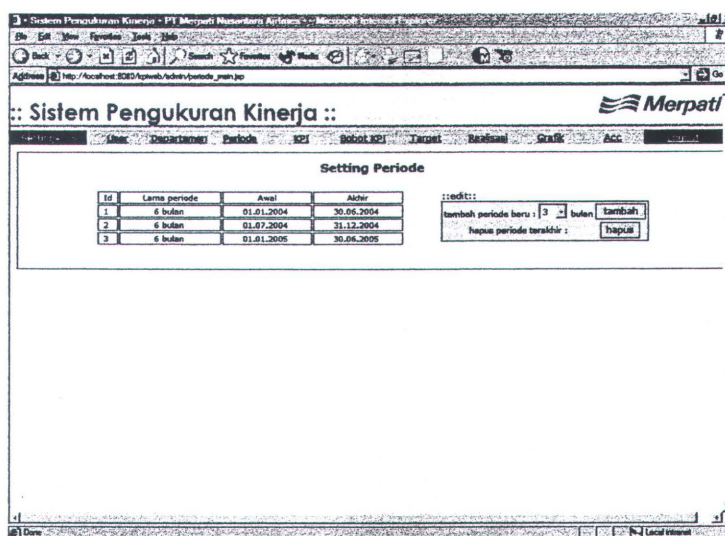
5.3.3. Implementasi Antarmuka Setting Periode

Implementasi antarmuka pada setting periode dalam bentuk file JSP pada file *periode_main.jsp* pada gambar 5.21 dan tampilan antarmuka gambar 5.22.

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="myBean" scope="page"
class="kamal.ta.beans.html.DBView"/>
<jsp:useBean id="authBean" scope="page"
class="kamal.ta.beans.controller.RoleBean"/>
<%
HttpSession mysession = request.getSession();
String sess_name = (String)mysession.getAttribute("name");
if(sess_name==null || "".equals(sess_name))
{ %>
<jsp:forward page="/depan.jsp?status=timeout"/>
<% }
if(!authBean.authByRole("Periode", sess_name))
{ %>-
<jsp:forward page="/accessdenied.jsp"/>
<% }

String Query = "select per_id as
Id, CONVERT (NVARCHAR(30), datediff(month, per_start, per_final)+1) + '
bulan' as 'Lama periode', CONVERT (NVARCHAR(30), per_start, 104) as
Awal, CONVERT (NVARCHAR(30), per_final, 104) as Akhir from tperiode";
myBean.setDataview(Query);
myBean.setCssTable("tabel1");
myBean.setCssRow("row1");
myBean.setCssHeader("row1");
myBean.setTableWidth("90%");
myBean.initTable();
%>
```

Gambar 5.21 Implementasi Antarmuka Setting Periode



Gambar 5.22 Tampilan Antarmuka Seting Periode

5.3.4. Implementasi Antarmuka Input dan Update KPI

Input dan update KPI diimplementasikan dengan antarmuka pada file *tree_edit_right.jsp* pada gambar 5.23. Yaitu untuk menampilkan anak (*child*) KPI yang diklik. Lalu untuk tampilan sebelah kiri atas dengan *tree_edit_node_info.jsp* pada gambar 5.24 untuk menampilkan info dari KPI yang diklik pada tree. Dan untuk sebelah bawah dengan *tree_edit_left.jsp* pada gambar 5.25 yang mengatur tampilan tree dan isi data tree dalam bentuk tabel.

Ketiga file JSP tersebut ditampilkan dalam satu halaman untuk input dan update KPI pada tampilan antarmuka gambar 5.26.

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.util.Vector,java.util.Iterator" errorPage="/error.jsp" %>
<!-- bean untuk listbox departemen-->
<jsp:useBean id="listBean" scope="page"
class="kamal.ta.beans.html.CostumListBox"/>
<jsp:setProperty name="listBean" property="query" value="select
dept_name, dept_id from tdept"/>
<!-- bean tree untuk ngambil data ttg node :treeBean.getNodeInfo()-->
<jsp:useBean id="treeBean" scope="session"
class="kamal.ta.beans.controller.KpiTreeBean"/>
<%
int numRecords = 0;
listBean.setName("c_kpi_dept_id");
String kpi_id = request.getParameter("selected");

boolean isRoot = false;
if(kpi_id==null)
    kpi_id = "0";
Vector[] val = treeBean.getNodeInfo(Integer.parseInt(kpi_id));
%>
```

Gambar 5.23 Implementasi Antarmuka *tree_edit_right.jsp*

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.util.Vector,java.util.Iterator" errorPage="/error.jsp" %>
<jsp:useBean id="listBean" scope="page"
class="kamal.ta.beans.html.CostumListBox"/>
<jsp:useBean id="hlmListBean" scope="page"
class="kamal.ta.beans.html.CostumListBox"/>
<jsp:setProperty name="listBean" property="query" value="select
dept_name, dept_id from tdept"/>
<jsp:setProperty name="hlmListBean" property="query" value="select
usr_id + ' - ' + usr_username , usr_id from tuser"/>
<jsp:useBean id="treeBean" scope="session"
class="kamal.ta.beans.controller.KpiTreeBean"/><%-- bean tree untuk
ngambil data ttg node :treeBean.getNodeInfo() --%>
<%
int numRecords = 0;
listBean.setName("c_kpi_dept_id");
String kpi_id = request.getParameter("selected");
boolean isRoot = false;
if(kpi_id==null)
    kpi_id = "0";
Vector[] val = treeBean.getNodeInfo(Integer.parseInt(kpi_id));
Object[] form1 = new Object[8];
//urut index array: [0]id kpi, [1]dept_id,
[2]tde_dept_id,[3]nama,[4]deskripsi, [5] unit, [6] root,[7]negate
String cb_negate=""; // untuk combo box negate
String strDisabled="";
if(val[0] != null){
    isRoot = false;
    form1 = val[0].toArray();
    if(form1[3]==null||"".equals(form1[3]))
        form1[3] = "---";
    if(form1[4]==null||"".equals(form1[4]))
        form1[4] = "---";
    if(form1[5]==null||"".equals(form1[5]))
        form1[5] = "---";
    if(form1[7]==null || "0".equals(form1[7].toString()) ){
        cb_negate = "";
    }else{
        cb_negate = "checked";
    }
}
//root node ...
}else {
    form1[0] = "0";
    form1[1] = "";
    form1[2] = "";
    form1[3] = "KPI Tree [node root]";
    form1[4] = "---";
    form1[5] = "---";
    strDisabled="disabled style=\"color:#dddddd\"";
}%>

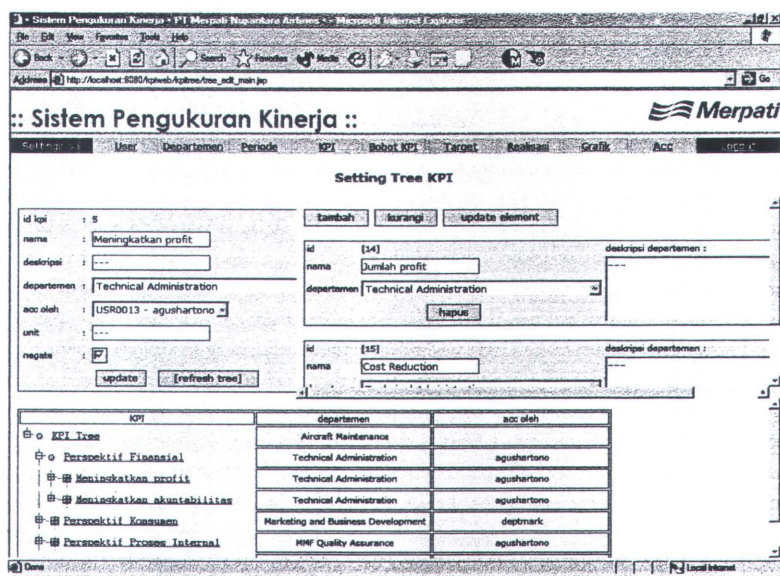
```

Gambar 5.24 Implementasi Antarmuka tree_edit_node_info.jsp

```
<%@ page import="kamal.ta.tagext.tree.impl.TreeNode,
               kamal.ta.tagext.tree.impl.Tree,
               kamal.ta.tagext.tree.itf.*, java.util.*"%>
<jsp:useBean id="treeBean" scope="session"
class="kamal.ta.beans.controller.KpiTreeBean"/>

<html>
<link href="../style/MatrixTour.css" type="text/css" rel="stylesheet"
/>
<body class="Normal">
<%@ taglib uri="/WEB-INF/treetag.tld" prefix="tree" %>
<%
String thisfile = "tree_edit_left.jsp";
String refresh = "";
refresh = request.getParameter("refresh");
if(session.getAttribute("example") == null ||
"ok".equals(refresh)) {
    treeBean.loadData();
    ITree kpiTree = treeBean.getTree();
    HashSet expandList = (HashSet)
session.getAttribute("expandedNodes");
    if(expandList!= null)
    {
        //out.print("expanding");
        Iterator iter = expandList.iterator();
        while(iter.hasNext())
        {
            ITreeNode node = (ITreeNode)iter.next();
            kpiTree.expand(node.getId());
            //out.println("[ "+node.getId()+" ] ");
        }
    }
    session.setAttribute("example", kpiTree);
}
%>
```

Gambar 5.25 Implementasi Antarmuka tree_edit_left.jsp



Gambar 5.26 Tampilan Antarmuka Input dan Update KPI

5.3.5. Implementasi Antarmuka Input dan Update Bobot KPI

Implementasi antarmuka *usecase* antarmuka input dan update bobot KPI *tree_weight_bottom.jsp* pada gambar 5.27 sebagai halaman utama sebelah atas. Dan untuk menampilkan semua KPI dalam bentuk tree beserta bobot-bobotnya, diimplementasikan dengan *tree_weight_top.jsp* pada gambar 5.29. Kedua file JSP tersebut ditampilkan dalam satu halaman untuk input dan update Bobot KPI pada tampilan antarmuka gambar 5.28.

```
<%
//parameter per_id diperoleh dari tree_target_bottom_act.jsp
String strPer_id = request.getParameter("per_id");
int iper_id = 1;
if(strPer_id != null)
    iper_id = Integer.parseInt(strPer_id);
periodBean.setIdSelected(iper_id-1);
%>
periode : <jsp:getProperty name="periodBean" property="html"/>
```

Gambar 5.27 Implementasi Antarmuka *tree_weight_bottom.jsp*

Sistem Pengukuran Kinerja :: Merpati

Settings >> User Departemen Periode KPI Bobot KPI Target Realisasi Grafik Acc Logout

Analytical Hierarchy Process

Indicator	Frekuensi Pelaporan	Current Ratio	Quick Ratio	Weight
Frekuensi Pelaporan	1	5	1	0.5051
Current Ratio	0.2	1	4	0.3551
Quick Ratio	1	0.25	1	0.1389
Total				0.9991

Cek Konsistensi

Consistency Ratio = 0,02

tree bobot kpi pada periode 1

id kpi	bobot	aktif
0	5.605	
1	1.999	
5	1.000	
6	0.999	
16	0.333	<input checked="" type="checkbox"/>
17	0.333	<input checked="" type="checkbox"/>
18	0.333	<input checked="" type="checkbox"/>
2	1.999	
3	0.608	

update

Gambar 5.28 Tampilan Antarmuka Input dan Update Bobot KPI

```

<%@ page import="kamal.ta.tagext.tree.impl.TreeNode,
               kamal.ta.tagext.tree.impl.Tree,
               kamal.ta.tagext.tree.itf.*,
               java.util.*"%>
<jsp:useBean id="treeBean" scope="session"
class="kamal.ta.beans.controller.KpiTreeBean"/>

<!-- window.top.app_details.document.location =
"frame_app_roles.jsp?" + url; -->
<html>
<link href="../style/MatrixTour.css" type="text/css" rel="stylesheet"
/>
<body class="Normal">
<%@ taglib uri="/WEB-INF/treetag.tld" prefix="tree" %>
<%
String thisfile = "tree_edit_left.jsp";
String refresh = "";
refresh = request.getParameter("refresh");
if(session.getAttribute("example") == null ||
"ok".equals(refresh)){
    treeBean.loadData();
    ITree kpiTree = treeBean.getTree();
    HashSet expandList = (HashSet)
session.getAttribute("expandedNodes");
    if(expandList!= null)
    {
        //out.print("expanding");
        Iterator iter = expandList.iterator();
        while(iter.hasNext())
        {
            ITreeNode node = (ITreeNode)iter.next();
            kpiTree.expand(node.getId());
            //out.println "["+node.getId()+" "];
        }
    }
    session.setAttribute("example", kpiTree);
}
%>

```

Gambar 5.29 Implementasi Antarmuka tree_weight_top.jsp

*"Tidak akan bergerak kaki seorang hamba pada hari kiamat,
sebelum ditanya,
tentang usia untuk apa dihabiskan,
tentang ilmu untuk apa dipergunakan,
tentang harta darimana diperoleh dan kemana dibelanjakan,
tentang jasad untuk apa digunakan"*
(HR. Tirmidzi)

BAB VI

UJI COBA

DAN EVALUASI PERANGKAT LUNAK

BAB VI

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Bab ini menjelaskan tentang tahap ujicoba dan evaluasi perangkat lunak. Pengujian pada perangkat lunak / aplikasi sistem pengukuran kinerja dapat dilakukan setelah proses perancangan dan implementasi selesai. Uji coba dilakukan untuk mengetahui apakah aplikasi yang telah dibuat dapat memecahkan permasalahan-permasalahan yang menjadi latar belakang dibuatnya aplikasi tersebut.

6.1. LINGKUNGAN UJI COBA

Lingkungan uji coba adalah komputer-komputer tempat dimana uji coba sistem dan aplikasi dilakukan. Tiap-tiap komputer mempunyai spesifikasi khusus agar uji coba dilakukan dengan lancar. Pada uji coba ini dibedakan menjadi dua bagian yaitu server dan client.

6.1.1. Server

Server yang digunakan untuk uji coba adalah Pentium 4 dengan kecepatan 2,4 GHz dan kapasitas DDRAM sebesar 448 Mb, kapasitas hard disk 30 Gb. Untuk spesifikasi minimum adalah server Pentium II dengan RAM sebesar 128 Mb.

Untuk spesifikasi software-nya adalah operating system Windows XP Professional, database Microsoft SQL Server 2000 sebagai tempat penyimpanan

data yang akan diuji coba, *web server Tomcat 4.1.27* untuk menangani halaman-halaman yang akan diakses oleh client, baik itu halaman JSP maupun HTML.

6.1.2. Client

Pada sisi client tidak ada spesifikasi hardware khusus. Yang diperlukan adalah komputer dengan *operating system windows* yang mendukung Internet Explorer 5.0 ke atas. Atau juga dapat menggunakan internet browser lainnya.

6.2. DATA UJI COBA

Uji coba menggunakan beberapa data awal, antara lain:

a. Data pengguna dan *password*

Tabel 6.1 Data Ujicoba Pengguna dan Password

No	Nama User	Password	Hak Akses
1	Admin	admin	Administrator
2	HRD	hrd	HRD
3	agushartono	agushartono	HLLM
4	sudirman	sudirman	MLLM

b. Data Periode

Tabel 6.2 Data Ujicoba Periode

No	Lama Periode	Tanggal Awal	Tanggal Akhir
1	6 bulan	01.01.2004	30.06.2004
2	6 bulan	01.07.2004	31.12.2004
3	6 bulan	01.01.2005	30.06.2005

c. Data KPI

Data uji coba KPI dapat dilihat pada subbab 3.1 dan 3.2 tentang KPI.

d. Data Matriks Perbandingan Berpasangan AHP

Data contoh kuisioner matriks berpasangan dapat dilihat pada lampiran A.

e. Data Target KPI

Tabel 6.3 Data Ujicoba Target

Indikator	Target	Satuan
F.1.1. Jumlah profit	1,64	Milyar
F.1.2. <i>Cost Reduction</i>	5	Persen/tahun
F.2.1. Frekuensi Pelaporan	80	Persen
F.2.2. <i>Current Ratio</i>	110	Persen
F.2.3. <i>Quick Ratio</i>	110	Persen
C.1.1. Kepuasan pelanggan	90	Persen
C.1.2. Prosentase <i>order</i> terkomplain	5	Persen
C.1.3. Nominal <i>order</i> terkomplain	3	Buah
C.2.1. Prosentase <i>market share</i> yang diraih	20	Persen
C.2.2. Pertumbuhan pelanggan	2	Pelanggan/tahun
I.1.1. <i>Post Inspection Defect</i>	5	Persen
I.1.2. Kualitas supplier	80	Persen
I.1.3. Jumlah kecelakaan kerja	1/1000	buah/peroperasi
I.2.1. <i>Turn around Time</i>	5	Persen
I.2.2. Kesalahan pembelian	5	Persen
I.2.3. Kesalahan pengiriman	5	Persen
I.3.1. <i>Component Shop</i>	10	Persen
I.3.2. <i>Engine & Propeller Shop</i>	10	Persen
I.3.3. <i>Heavy Maintenance</i>	10	persen
L.1.1. <i>Employee Training Hours</i>	40	Jam/tahun
L.1.2. <i>Labour Target Achievement</i>	90	Persen
L.1.3. Kesejahteraan karyawan	20	Persen/tahun
L.1.4. Jumlah karyawan dg keluhan kerja	0	Pegawai
L.2.1. Rasio Jumlah komputer dg pegawai	1/5	Komputerpegawai
L.2.2. Jumlah klaim sistem informasi	3	Buah

f. Data Pencapaian KPI

Tabel 6.4 Data Ujicoba Pencapaian

Indikator	Bulan 1-2	Bulan 3-4	Bulan 5-6	Satuan
F.1.1. Jumlah profit	0,5	0,51	0,3	Milyar
F.1.2. <i>Cost Reduction</i>	5	4,5	5	Persen/tahun
F.2.1. Frekuensi Pelaporan	80	75	70	Persen

Indikator	Bulan 1-2	Bulan 3-4	Bulan 5-6	Satuan
F.2.2. <i>Current Ratio</i>	110	105	100	Persen
F.2.3. <i>Quick Ratio</i>	110	110	107	Persen
C.1.1. Kepuasan pelanggan	90	85	92	Persen
C.1.2. Prosentase <i>order</i> terkomplain	5	4	10	Persen
C.1.3. Nominal <i>order</i> terkomplain	5	4	8	Buah
C.2.1. Prosentase <i>market share</i> yang diraih	20	15	14	Persen
C.2.2. Pertumbuhan pelanggan	2	2	0	Pelanggan/tahun
I.1.1. <i>Post Inspection Defect</i>	5	5	3	Persen
I.1.2. Kualitas supplier	80	90	90	Persen
I.1.3. Jumlah kecelakaan kerja	1/1000	1/1500	0	buah/peroperasi
I.2.1. <i>Turn around Time</i>	5	4	5	Persen
I.2.2. Kesalahan pembelian	5	3	1	Persen
I.2.3. Kesalahan pengiriman	5	5	2	Persen
I.3.1. <i>Component Shop</i>	10	7	7	Persen
I.3.2. <i>Engine & Propeller Shop</i>	10	17	15	Persen
I.3.3. <i>Heavy Maintenance</i>	10	5	5	persen
L.1.1. <i>Employee Training Hours</i>	40	25	20	Jam/tahun
L.1.2. <i>Labour Target Achievement</i>	90	85	80	Persen
L.1.3. Kesejahteraan karyawan	20	20	18	Persen/tahun
L.1.4. Jumlah karyawan dg keluhan kerja	0	1	0	Pegawai
L.2.1. Rasio Jumlah komputer dg pegawai	1/5	1/5	1/5	Komputerpegawai
L.2.2. Jumlah klaim sistem informasi	5	3	0	Buah

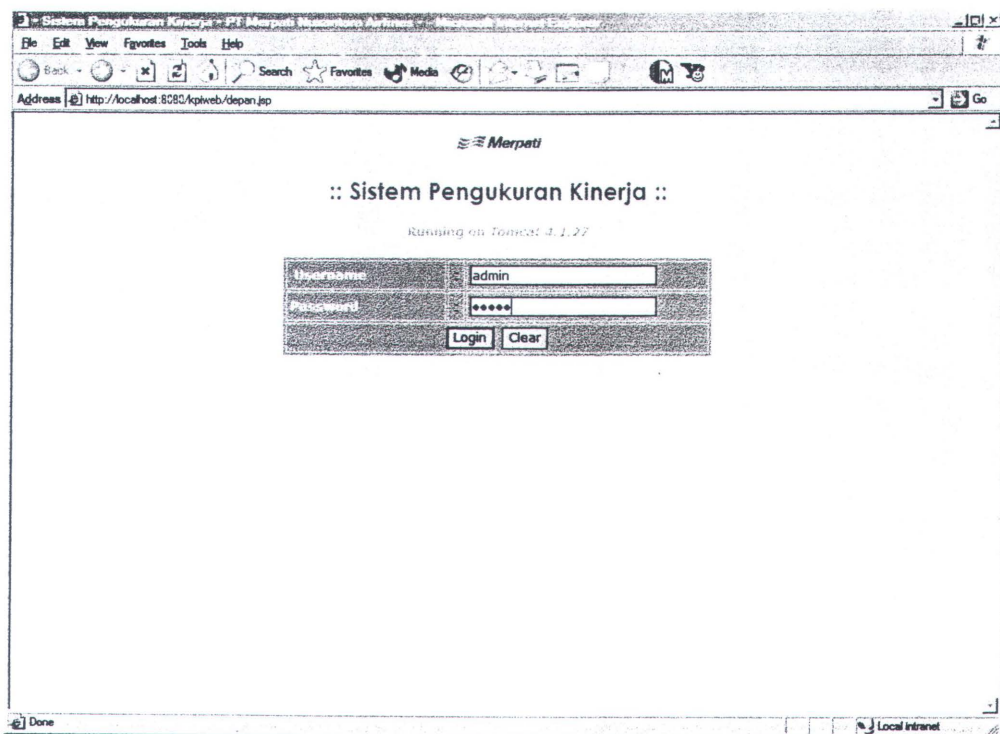
6.3. PELAKSANAAN UJI COBA

Uji coba dilakukan dengan empat buah skenario yang berbeda. Uji coba dibedakan berdasarkan jenis aktor yang terlibat dalam sistem, yaitu skenario pertama bagi administrator, skenario kedua bagi HRD, skenario ketiga bagi HLM dan skenario keempat bagi MLLM.

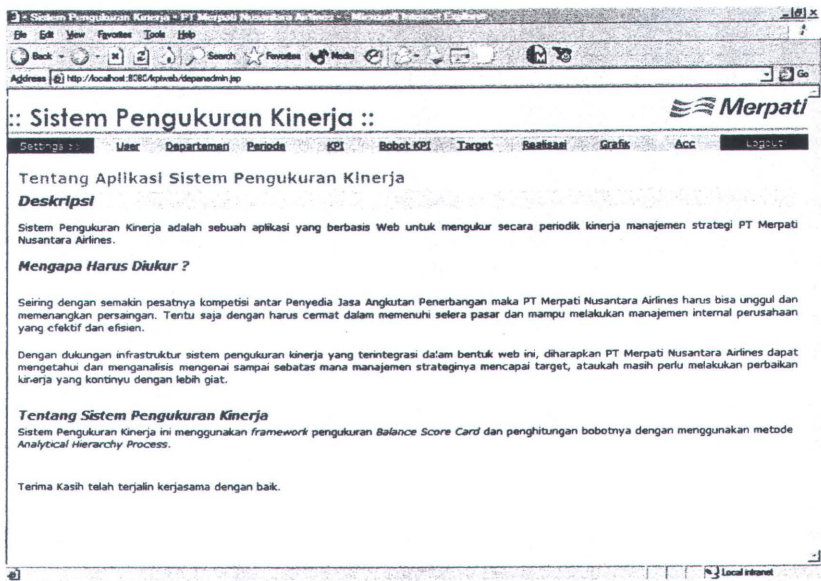
6.2.1. Skenario Pertama Sebagai Aktor Administrator

Uji coba yang pertama ini dilakukan oleh login dengan nama pengguna *admin* dengan hak akses sebagai administrator. Pengguna *admin* akan melakukan uji coba sesuai dengan hak aksesnya.

Pertama kali, sebelum masuk ke sistem, pengguna harus melakukan login sebagai administrator. Ketika pengguna *admin* login ke sistem (gambar 6.1), maka secara otomatis, sistem akan mengecek apakah jenis hak akses dari pengguna tersebut. Dan sesuai dengan database, *admin* tercatat mempunyai hak akses sebagai administrator. Setelah login terlewati, pengguna akan memasuki halaman utama aplikasi pada gambar 6.2.

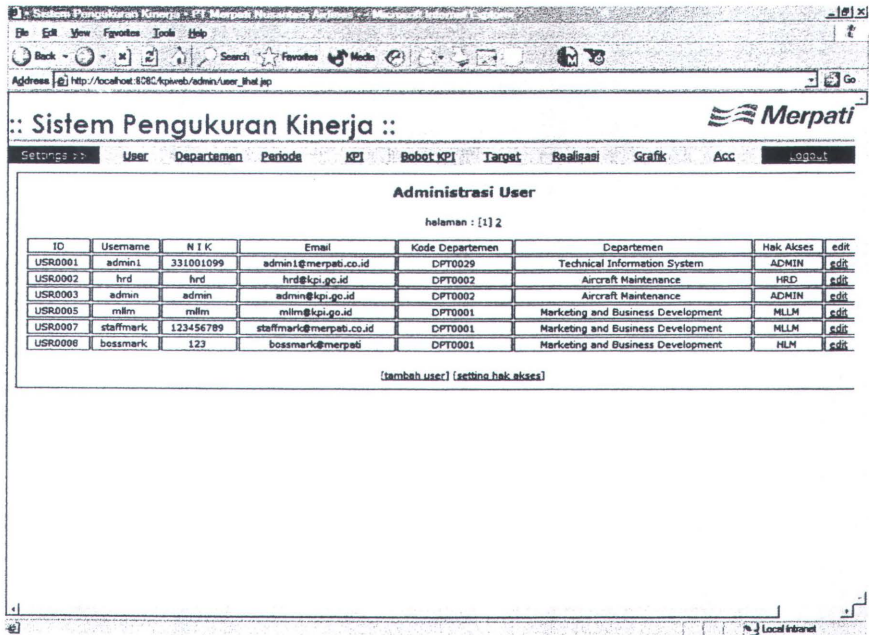


Gambar 6.1 Login Pengguna admin sebagai Aktor Administrator

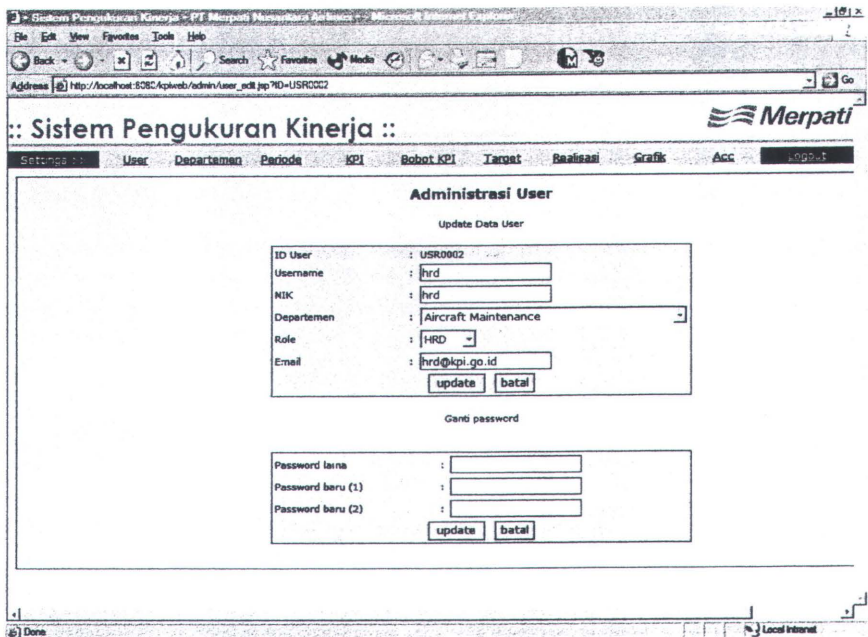


Gambar 6.2 Halaman Utama Aplikasi

Salah satu hak akses yang hanya dimiliki oleh aktor administrator adalah *maintenance* pengguna *dan* *role* pada gambar 6.3. Pengguna *admin* akan melakukan uji coba dengan menambah pengguna baru, mengedit pengguna lama *dan* mengganti *password* pengguna.

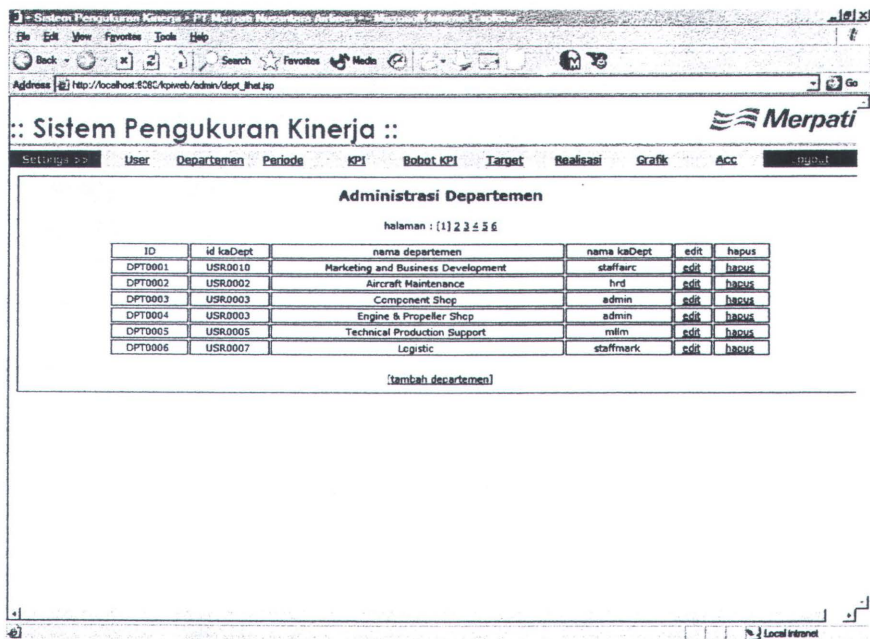


Gambar 6.3 Manage Pengguna dan Role

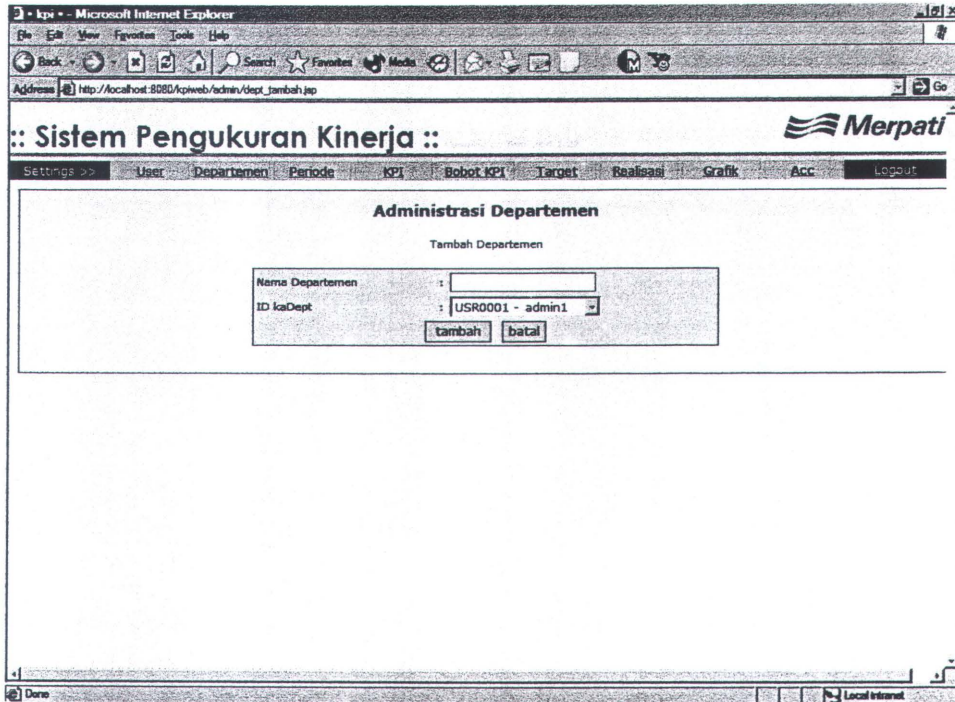


Gambar 6.4 Edit Data Pengguna dan Password

Selanjutnya adalah uji coba untuk *manage* departemen. Pengguna *admin* akan melakukan uji coba menambah departemen baru pada gambar 6.5, mengedit data departemen lama dan menghapus data departemen lama pada gambar 6.6.



Gambar 6.5 Manage Departemen



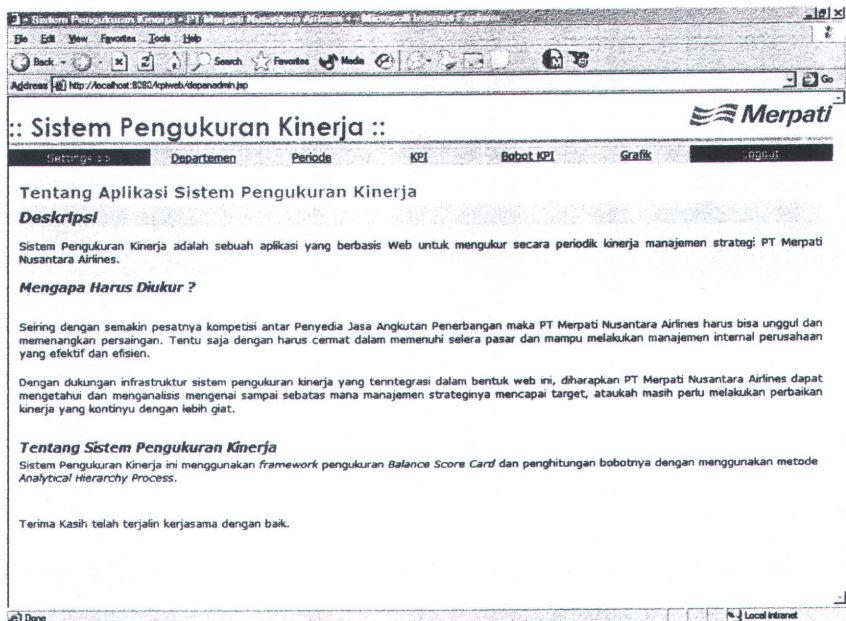
Gambar 6.6 Tambah Departemen Baru

6.2.2. Skenario Kedua Sebagai HRD

Uji coba yang kedua dilakukan oleh login dengan *username hrd* dengan hak akses sebagai HRD. Pengguna *hrd* akan melakukan uji coba sesuai dengan hak aksesnya.

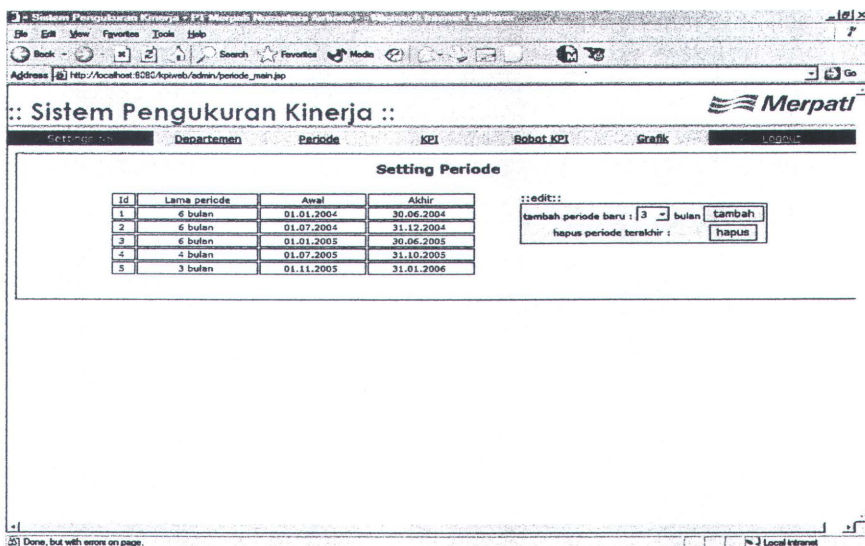
Asumsi pengguna *hrd* sudah login sebagai aktor HRD. Dan selanjutnya, sesuai dengan database, *hrd* tercatat mempunyai hak akses sebagai administrator. Setelah login terlewati, pengguna akan memasuki halaman utama aplikasi.

Dari gambar 6.7 dapat terlihat bahwa, menu yang dimiliki oleh pengguna *hrd* sebagai aktor HRD adalah sesuai dengan hak aksesnya. Menu yang bukan hak akses aktor HRD tidak akan ditampilkan.



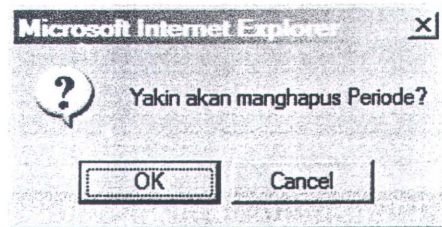
Gambar 6.7 Halaman Utama Login hrd sebagai aktor HRD

Selanjutnya pengguna melakukan uji coba untuk menyetting periode pengukuran kinerja seperti pada gambar 6.8. Pengguna dapat menambah periode baru pengukuran kinerja dan juga menghapus periode terakhir yang tidak diperlukan.



Gambar 6.8 Setting Periode

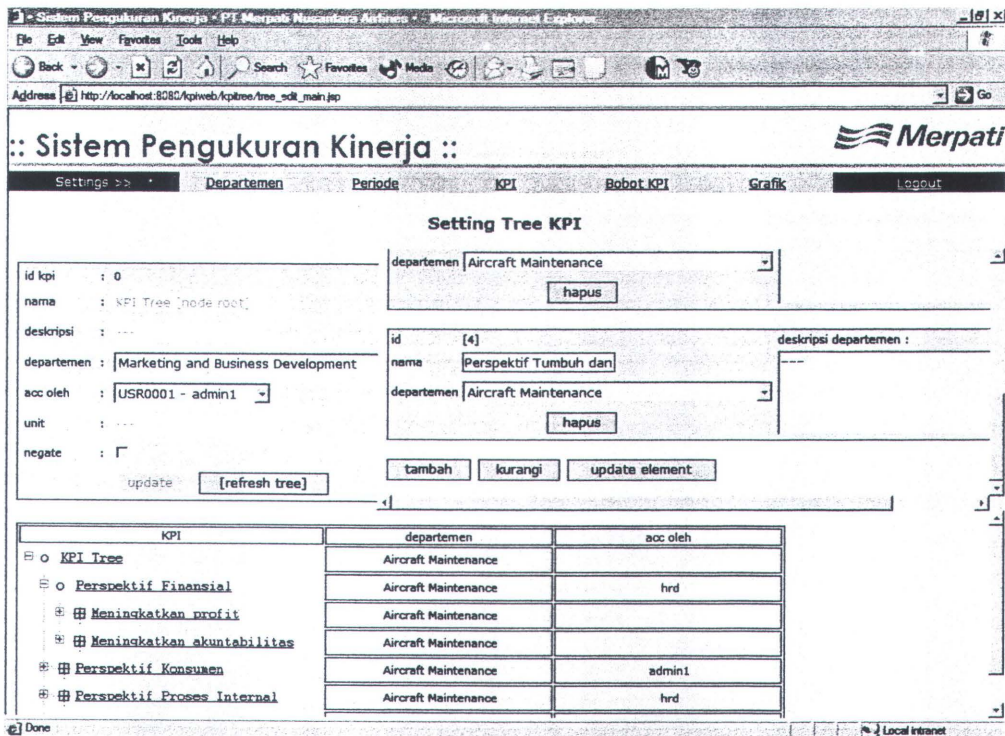
Jika pengguna akan menghapus periode, maka akan ada konfirmasi apakah pengguna yakin akan menghapus periode ataukah tidak, seperti tampak pada gambar 6.9.



Gambar 6.9 Konfirmasi Menghapus Periode

Uji coba selanjutnya adalah pengguna melakukan *input dan update KPI*. Input dilakukan dengan mengklik pada KPI pada tree yang menjadi induk (*parent*) dari KPI yang akan diinputkan.

Setelah itu seperti terlihat pada gambar 6.10, pengguna akan memasukkan KPI baru dengan mengklik pada tombol tambah sebelah kanan atas.

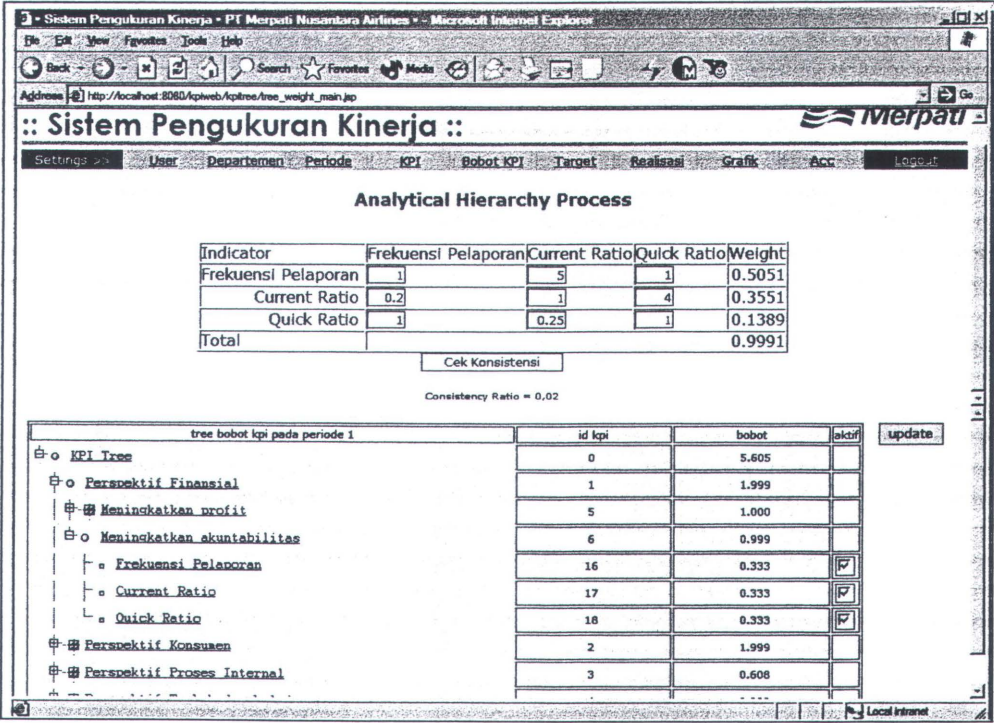


Gambar 6.10 Input dan Update KPI

Setiap kali melakukan input atau update pada KPI, jangan lupa untuk menekan [update element] atau [Refresh Tree] agar tree yang menampilkan KPI dapat ter-update.

Untuk semua data KPI ini, akan ditampilkan semuanya sesuai dengan yang ada dalam database. Dan jika akan digunakan, maka harus melakukan setting KPI Aktif Periodik terlebih dahulu sebelum KPI diberi bobot setiap periodenya.

Uji Coba selanjutnya adalah melakukan input dan update bobot KPI. Pada *usecase* ini, harus melakukan setting KPI Aktif Periodik terlebih dahulu, sesuai dengan yang telah digambarkan pada *usecase* realisasi, karena Setting KPI Aktif Periodik *include* ke dalam input dan update bobot KPI. Untuk melakukan penghitungan bobot maka digunakan metode AHP dengan mengisi pada matriks berpasangan. Selengkapny dapat dilihat pada gambar 6.11.

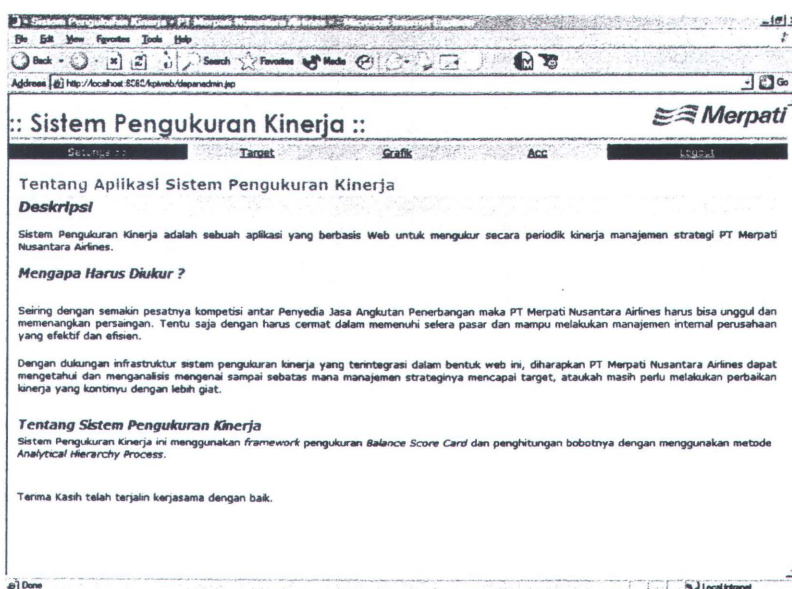


Gambar 6.11 Input dan Update Bobot KPI dengan metode AHP

6.2.3. Skenario Ketiga Sebagai HLM

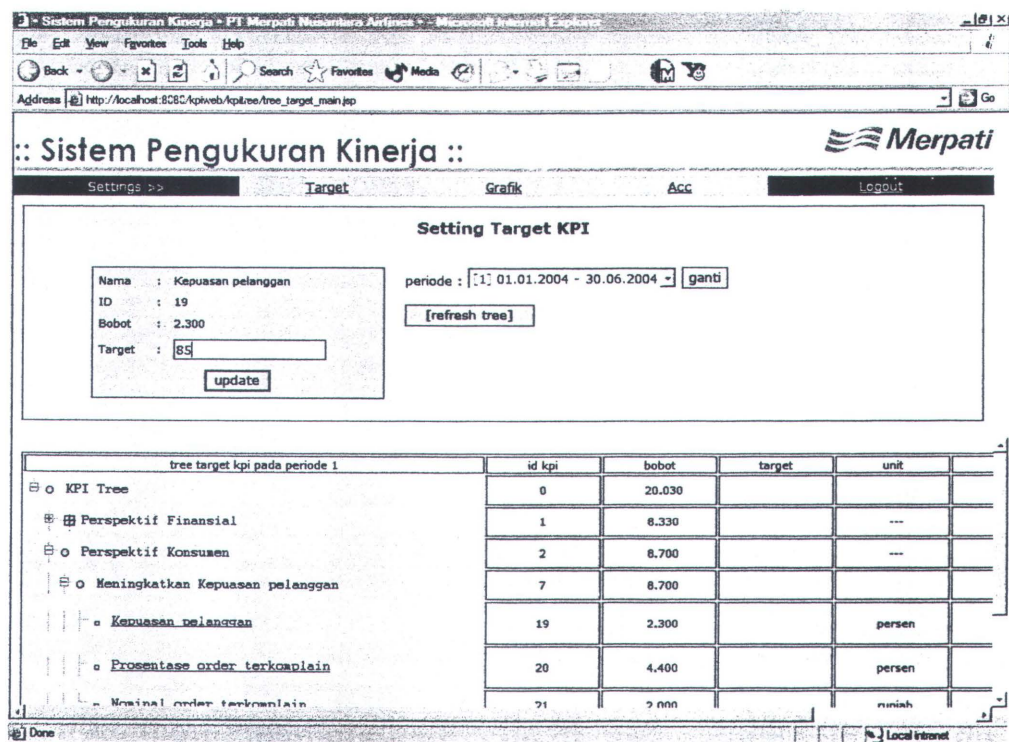
Uji coba ketiga adalah dengan *username agushartono* sebagai HLM, yaitu sebagai wakil dari general manager marketing. Asumsi bahwa login telah selesai dilaksanakan. Selanjutnya langsung pada halaman utama HLM, dapat dilihat pada gambar 6.13.

Usecase yang hanya dimiliki oleh HLM adalah *menentukan target KPI*. Pada *usecase*, sistem akan secara otomatis mengecek *username* dari pengguna. Jika *username* sama dengan pengguna yang berhak memasukkan target bagi suatu KPI, dapat dilihat pada gambar 6.14, maka sistem akan secara otomatis dapat memberikan hak edit bagi pengguna.



Gambar 6.12 Halaman Utama HLM

Jika bukan pengguna yang bersangkutan maka sistem tidak akan memberikan hak untuk menentukan target KPI, walaupun memiliki hak akses sebagai HLM.



Gambar 6.13 Menentukan Target KPI Periodik

6.2.4. Skenario Keempat Sebagai MLLM

Skenario keempat adalah pengguna *sudirman* login sebagai aktor MLLM dengan departemen pada marketing and business development department. Satu-satunya hak akses yang hanya dimiliki oleh MLLM adalah *melakukan pelaporan pencapaian KPI periodik*, seperti dapat dilihat pada gambar 6.15. Pengguna melakukan pelaporan hanya pada departemen yang bersangkutan. Tidak dapat melakukan pelaporan bagi KPI milik departemen lain.

http://localhost:8080/kpiweb/kpitree/tree_perform_main.jsp - Microsoft Internet Explorer

Address: http://localhost:8080/kpiweb/kpitree/tree_perform_main.jsp

:: Sistem Pengukuran Kinerja :: Running on JSP 4.0.27

Merpati

Settings >> Realisasi Grafik Logout

Setting Realisasi KPI

Name : Kepuasan pelanggan

ID : 19

Bobot : 2.300

Target : ---

Realisasi : ---

update

periode : [1] 01.01.2004 - 30.06.2004

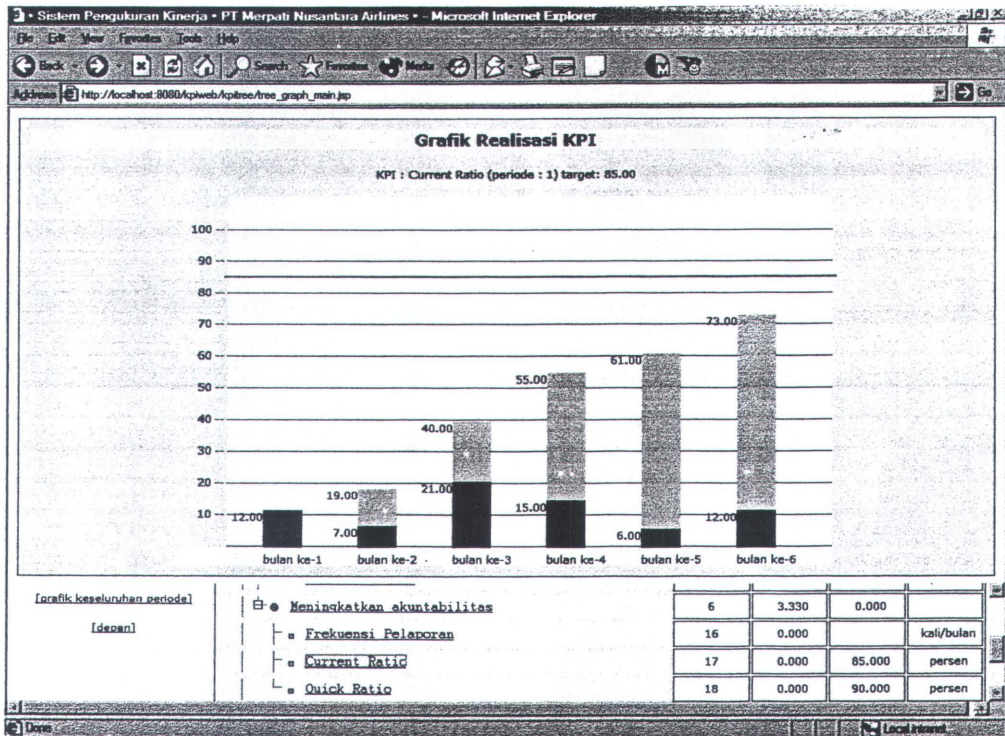
bulan ke : [1] ganti

[refresh tree]

tree realisasi kpi pada periode 1 bulan ke 1		id kpi	bobot	target	realisasi	unit
KPI Tree		0	20.030			
Perspektif Finansial		1	8.330			
Perspektif Konsumen		2	8.700			
Meningkatkan Kepuasan pelanggan		7	8.700			
Kepuasan pelanggan		19	2.300			persen
Prosentase order terkomplain		20	4.400			persen
Nominal order terkomplain		21	2.000			rupiah
Meningkatkan market share		8				

Gambar 6.14 Melaporkan pencapaian KPI periodik

Pencapaian KPI dilaporkan setiap bulannya selama masa periode pengukuran kinerja. Jadi misalnya seperti pada uji coba, bahwa periode pengukuran kinerja diset selama enam bulan. Maka aktor MLLM melakukan pelaporan pencapaian enam kali dalam satu periode. Uji coba terakhir, semua pengguna dapat melakukan pengukuran kinerja setiap periodenya. Pengguna mengklik pada KPI yang ingindiketahui kinerjanya pada suatu periode. Misalnya KPI *current ratio*. Seperti terlihat pada gambar 6.16.



Gambar 6.15 Melakukan Pengukuran Kinerja

EVALUASI UJI COBA

Setelah melakukan uji coba, perlu dilakukan evaluasi terhadap hasil uji coba tersebut. Hasil evaluasi uji coba perangkat lunak adalah sebagai berikut:

- Pada ujicoba sebagai aktor administrator, awal proses login sebagai role administrator berjalan dengan sukses. *Usecase* manage pengguna dan *role* berhasil. *Usecase* manage departemen berhasil. *Usecase* setting periode berhasil. Begitu juga dengan *usecase-usecase* lain.
- Pada ujicoba sebagai aktor HRD, awal proses login sebagai *role* HRD berjalan dengan sukses. *Usecase* manage departemen berhasil. *Usecase* setting periode berhasil. Begitu juga dengan *usecase-usecase* lain.

- c. Berdasarkan ujicoba sebagai aktor HRD, perhitungan pembobotan KPI dapat dilakukan dengan tepat dengan menggunakan metode AHP.
- d. Pada ujicoba sebagai aktor HLM, awal proses login sebagai *role* HLM berjalan dengan sukses. *Usecase* menentukan target KPI berhasil. *Usecase Accountability* berhasil. *Usecase* lihat hasil pengukuran kinerja juga berhasil.
- e. Pada ujicoba sebagai aktor MLLM, awal proses login sebagai *role* MLLM berjalan dengan sukses. *Usecase* melaporkan pencapaian KPI periodik berhasil. *Usecase* lihat hasil pengukuran kinerja juga berhasil.



Qum, Fa andzir...!!!


BAB VII
SIMPULAN

BAB VII

SIMPULAN

Beberapa simpulan yang dapat diambil dari proses pembuatan aplikasi Sistem Pengukuran Kinerja dalam Tugas Akhir ini adalah :

1. Berdasarkan hasil ujicoba terhadap keempat jenis skenario uji, perangkat lunak pengukuran kinerja ini telah berfungsi dengan benar sesuai dengan yang diinginkan.
2. Berdasarkan hasil uji coba perangkat lunak, simulasi pencapaian target sasaran di PT Merpati Nusantara Airlines Surabaya akan dapat dilakukan dengan mudah, cepat dan akurat. Dengan demikian, pihak manajemen dapat melakukan tindakan koreksi, yang antara lain dapat berupa kebijakan manajemen strategis untuk periode berikutnya guna meningkatkan daya saing perusahaan.



*Di dalam kegelapan
ku mencari cahaya-Mu
yang hilang sirna
tak tersisa*

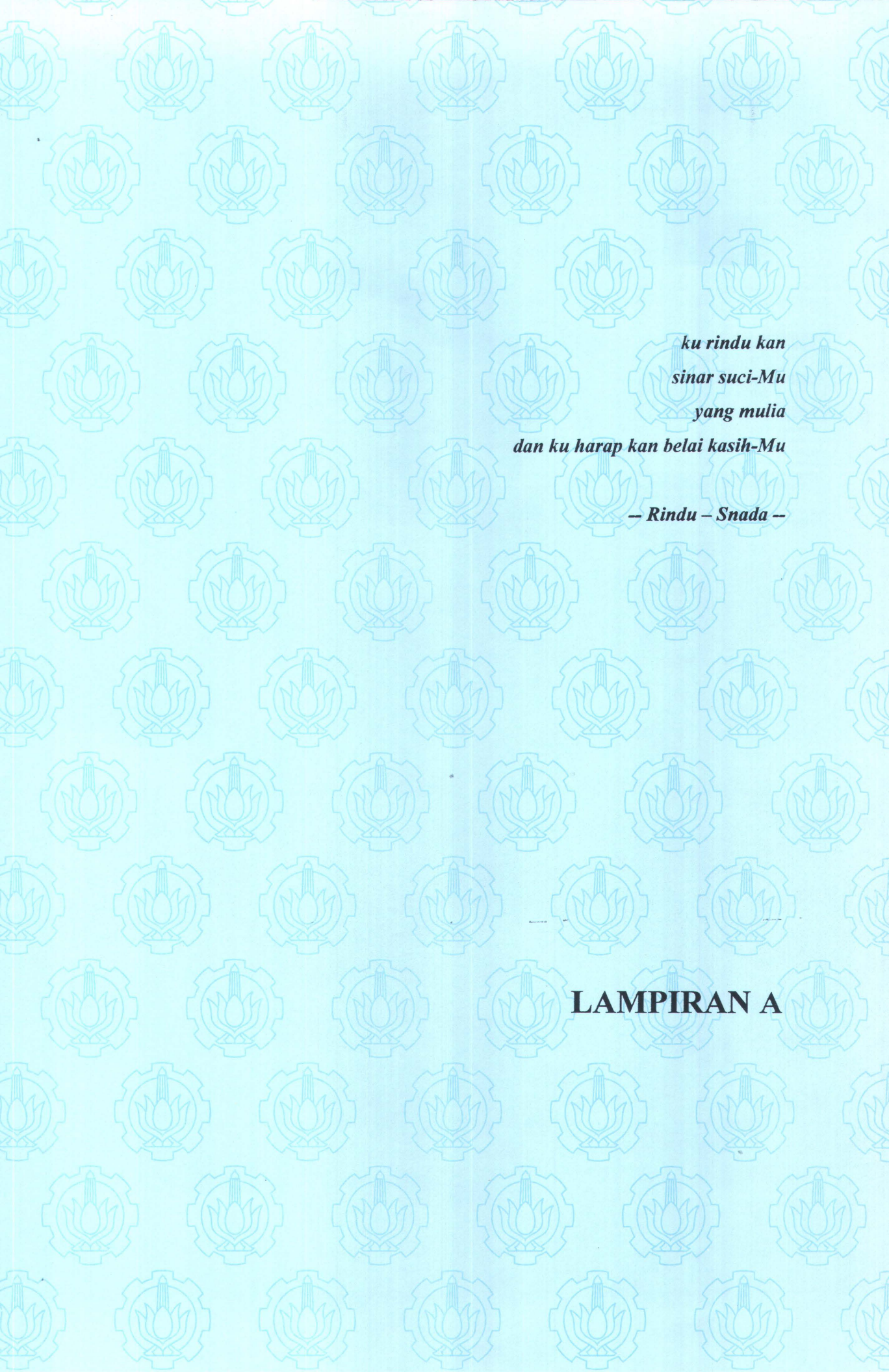
■ *Rindu – Snada –*

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [ARI04] Arief, M Fahmi, "Aplikasi QMPMS Pada Perancangan Sistem Pengukuran Kinerja SBU, Merpati Maintenance Facility", *Tugas Akhir*, Jurusan Teknik Industri-ITS, Surabaya, 2004.
- [ART01] Artley, Will, et.al., "The Performance Management Handbook : Establishing an Integrated Performance Measurement System", *Performance Based Management Special Interest Group (PBM SIG)*, <http://www.ora.gov/pbm>, 2001.
- [BRO02] Brown, David William, *An Introduction to Object-Oriented Analysis*, Second Edition, Northen Alberta Institute of Technology. 2002.
- [CAL98] Caliskan, DR Nurbanu, "A Decision Support Approach For The Evaluation Of Transport Investment Alternatives", *Technical Report University of Istanbul, Dept. of Faculty of Civil Engineering*, Transportation, www.iasi.rm.cnr.it/ewgt/13conference/141_caliskan.pdf, 1998.
- [HOF99] Hoffer, Jeffrey, *A Modern System Analysis dan Design*, 2nd edition, Mc. Graw-Hill, USA, 1999.
- [IND02] Indah, Luck, dkk., "Perancangan Sistem Pengukuran Kinerja dengan Balanced Scorecard pada PT Industri sandang Nusantara", *Proceeding Seminar Nasional TIMP 2002*, Jurusan Teknik Industri ITS, 2002.
- [QUA98] Quatrani, Terry, "Visual Modeling With Rational Rose And UML", Addison Wesley Longman, Massachusetts, 1998.
- [SAA80] Saaty, T.L, *The Analitical Hierarchy Process*, McGraw-Hill, New York, 1980.

- [SUH02] Suhendar, A, *Visual Modelling Menggunakan UML dan Rational Rose*, Penerbit Informatika Bandung, 2002.
- [SUM03] Sumbogo, Kendy Aditya, Aby Herwendo, “Pembuatan Aplikasi KPI (Key Performance Indicator) Online menggunakan Balanced Scorecard pada PT Indosat Tbk”, *Laporan Kerja Praktek*, Jurusan Teknik Informatika-ITS, Surabaya, 2003.
- [SUW03] Suwignjo, Patdono, dkk., “Perancangan Sistem Pengukuran Kinerja Yayasan Masjid Al Falah Unit Bisnis YDSF”, *Laporan Penelitian*, Lembaga Penelitian ITS, Surabaya, 2003.
- [VAN03] Vanany, Iwan, “Perancangan Sistem Pengukuran Kinerja Pada Perguruan Tinggi Dengan Menggunakan Metode Balanced Scorecard (Studi Kasus di Jurusan Teknik Industri-ITS Surabaya)”, *Tesis*, Jurusan Teknik Industri-ITS, Surabaya, 1999.



*ku rindu kan
sinar suci-Mu
yang mulia
dan ku harap kan belai kasih-Mu*

— Rindu — Snada —

LAMPIRAN A

LAMPIRAN A

KUISIONER PEMBOBOTAN KPI

Tujuan dari kuisisioner ini adalah untuk mengetahui bobot dari masing-masing KPI-KPI dari sistem pengukuran kinerja pada Merpati Maintenance Facility, nilai dari KPI-KPI adalah sebagai berikut :

Tabel A.1 Nilai Intensitas Kepentingan

Nilai	Definisi
1	Bobot kepentingan elemen keputusan yang satu dinilai sama penting dibandingkan elemen keputusan yang lain.
3	Bobot kepentingan elemen keputusan yang satu dinilai sedikit lebih penting dibandingkan elemen keputusan yang lain.
5	Bobot kepentingan elemen keputusan yang satu dinilai cukup penting dibandingkan elemen keputusan yang lain.
7	Bobot kepentingan elemen keputusan yang satu dinilai sangat penting dibandingkan elemen keputusan yang lain.
9	Bobot kepentingan elemen keputusan yang satu dinilai mutlak (sangat penting sekali) dibandingkan elemen keputusan yang lain.
2,4,6,8	Nilai tengah. Diberikan bila terdapat keraguan penilaian antara dua penilaian yang berdekatan.

Contoh:

Tabel A.2 Contoh Tabel Isian Matriks

	KPI 2	KPI 3	KPI 4
KPI 2			
KPI 3			
KPI 4			

Jika menurut anda KPI 2 lebih penting dari KPI 3, KPI 4 mutlak penting dari KPI 2 dan KPI 3 lebih sedikit penting dari KPI 4, maka hasil pengisian kuisisioner pada Tabel A.3 berikut.

Tabel A.3 Contoh Hasil Pengisian Matriks

	KPI 2	KPI 3	KPI 4
KPI 2		5	1/7
KPI 3			3
KPI 4			

Petunjuk: Silahkan diisi sesuai dengan contoh diatas.

Pembobotan Overall

Tabel A.4 Tabel Isian Overall

	Finansial	Konsumen	Proses Bisnis Internal	Proses Tumbuh dan Belajar
Finansial				
Konsumen				
Proses Bisnis Internal				
Proses Tumbuh dan Belajar				

Perspektif Konsumen

Tabel A.5 Tabel Isian Perspektif Konsumen

	Kepuasan Pelanggan	Market Share
Kepuasan Pelanggan		
Market share		

Kriteria Kepuasan Pelanggan

Tabel A.6 Tabel Isian Kriteria Kepuasan Pelanggan

	Kepuasan Pelanggan	Prosentase Order terklaim	Nominal Order terklaim
Kepuasan Pelanggan			
Prosentase Order terklaim			
Nominal Order terklaim			

Kriteria Market Share

Tabel A.7 Tabel Isian Kriteria Market Share

	Prosentase market share yang diraih	Pertumbuhan Pelanggan
Prosentase market share diraih		
Pertumbuhan pelanggan		

Perspektif Finansial

Tabel A.8 Tabel Isian Perspektif Finansial

	Meningkatkan Akuntabilitas Kinerja Keuangan	Profit
Meningkatkan Akuntabilitas Kinerja Keuangan		
Profit		

Kriteria Profit

Tabel A.9 Tabel Isian Kriteria Profit

	Peningkatan Pendapatan	Menurunkan biaya operasi
Peningkatan Pendapatan		
Menurunkan biaya operasi		

Kriteria Meningkatkan Akuntabilitas Keuangan

Tabel A.10 Tabel Isian Kriteria Meningkatkan Akuntabilitas Keuangan

	Current Ratio	Quick Ratio	Frekuensi pelaporan
Current Ratio			
Quick Ratio			
Frekuensi pelaporan			

Perspektif Proses Bisnis Internal

Tabel A.11 Tabel Isian Perspektif Proses Bisnis Internal

	Kualitas	Produktivitas
Kualitas		
Produktivitas		

Kriteria Kualitas

Tabel A.12 Tabel Isian Kriteria Kualitas

	Post Infection Defect	Jumlah Kecelakaan	Kualitas Supplier
Post Infection Defect			
Jumlah kecelakaan			
Kualitas Supplier			



Kriteria Produktivitas

Tabel A.13 Tabel Isian Kriteria Produktivitas

	Kesalahan Pembelian	Kesalahan Pengiriman	Turn Around Time	Component shop	Engine & propeller shop	Heavy maintenance
Kesalahan Pembelian						
Kesalahan Pengiriman						
Turn Around Time						
Component shop						
Engine & propeller shop						
Heavy maintenance						

Perspektif Proses Tumbuh dan Belajar

Tabel A.14 Tabel Isian Perspektif Proses tumbuh dan Belajar

	SDM	Sistem Informasi
SDM		
Sistem Informasi		

Kriteria SDM

Tabel A.15 Tabel Isian Kriteria SDM

	Employee Training Hours	Prosentase pencapaian target karyawan	Kesejahteraan Karyawan	Jumlah karyawan dengan keluhan
Employee Training Hours				
Prosentase pencapaian target karyawan				
Kesejahteraan Karyawan				
Jumlah karyawan dengan keluhan				

Kriteria Sistem Informasi

Tabel A.16 Tabel Isian Kriteria Sistem Informasi

	Rasio jumlah komputer dengan karyawan	Jumlah klaim sistem informasi
Rasio jumlah komputer dengan karyawan		
Jumlah klaim sistem informasi		

Mengetahui
General Manager SBU MMF

Yunus Dzulisnain